



A Region-Selective Anti-compression Image Encryption Algorithm Based on Deep Networks

Zhenqiang Chen¹ · Yifeng Liu¹ · Gang Ke¹ · Jingkai Wang¹ · Weibin Zhao² · Sio-long Lo¹ 

Received: 19 January 2024 / Accepted: 21 April 2024
© The Author(s) 2024

Abstract

In recent years, related research has focused on how to safely transfer and protect the privacy of images in social network services while providing easy access by authorized users. To safeguard privacy, we suggest an image encryption scheme that combines data hiding and image encryption. The proposed scheme successfully decrypts images after JPEG compression attacks and preserves the privacy of secret regions through the use of block scrambling encryption based on region selection. Simultaneously, the scheme can handle nonuniform secret regions and obtain more sensitive secret keys because of the incorporation of a chaotic system. The enhanced deep learning-based data-hiding technology reduces algorithm complexity by enabling the encryption position to be determined in the decryption phase without the need for any information or equipment. However, this approach also increases algorithm security, because only when the right secret data are extracted can they be decrypted successfully. According to the experimental findings, the proposed scheme can correctly decrypt images via JPEG compression while maintaining visually acceptable quality. The proposed scheme can achieve greater robustness against image processing algorithms and a wider secret key space than traditional schemes.

Keywords JPEG compression · Region selection · Data hiding · Block scrambling · Deep learning

1 Introduction

Advances in cloud computing allow customers to use nearly infinite computational resources without incurring excessive expenditures (such as buying and maintaining extra hardware and software) [1]. Millions of people post images to the Inter-

net for sharing and archiving, and social network services (SNSs), such as Instagram and Twitter, have grown in popularity. Service providers usually preprocess uploaded images, such as by recompressing them, to lower the quantity of data that need to be processed while preserving the high quality of the images to minimize the expense of billions of messages. However, historically, many individuals, including social scientists and attorneys, have seen privacy as a social and legal concern [2]. Stored images are also susceptible to assaults (such as brute-force attacks and ciphertext-only attacks) and exploitation by unauthorized users, because SNSs are public or semipublic spaces [3]. For instance, once an image is uploaded to an SNS, control over the image is given to the SNS provider, whose reliability is unknown, increasing the likelihood of image data leakage [4]. Preventing unauthorized users from viewing and using images is essential for safeguarding personal privacy, since most images include privacy components such as copyright protection and personal data. Three requirements must be met to guarantee the security of images shared on social networks: (1) The image's visual information is adequately safeguarded. (2) The image can withstand recompression. (3) The image can be accu-

✉ Sio-long Lo
slllo@must.edu.mo

Zhenqiang Chen
3220002509@student.must.edu.mo

Yifeng Liu
3220005660@student.must.edu.mo

Gang Ke
kegang95@126.com

Jingkai Wang
3230002398@student.must.edu.mo

Weibin Zhao
6543265@qq.com

¹ Faculty of Innovation Engineering, Macau University of Science and Technology, Macao, China

² Department of Network Information Security, Guangdong Police College, Guangzhou, China

rately identified. However, ensuring all three requirements at once is typically feasible only in some schemes [5–7].

For requirement 1, using a secret key to transform the original image into one that looks like noise is known as image encryption, which is the most straightforward and efficient method of ensuring data security. The image is encrypted using confusion and diffusion, and without the correct secret key, the user cannot decrypt it. Numerous academics have proposed different image encryption techniques, including the chaotic system, the Advanced Encryption Standard (AES), and the Data Encryption Standard (DES) [1, 4–11]. Since chaotic systems exhibit good random behavior, sensitivity to the initial value, and unpredictability, they are commonly employed in image encryption. However, because of the volume of data and the continuity among image pixels, the conventional encryption technique in the image encryption scheme is under threat. Ensuring security necessitates a trade-off with other needs, such as minimization of processor requirements, bitstream regulatory compliance, and cryptographic signal processing [9]. Hu et al. proposed a low-complexity pseudorandom number generation algorithm that satisfied the requirements of image encryption by utilizing the hyperchaotic properties of cellular neural networks [12]. Using the secret key produced by prime numbers, Indira et al. suggested a lightweight active padding encryption scheme that can lower the time complexity [13]. Abdullah et al. first transform an RGB image's 3D matrix to obtain the optimal entropy and contrast criterion for image encryption [14].

For requirement 2, the most widely used lossy compression operation for digital images is DCT-based JPEG compression, which aims to decrease capacity, enhance the transmission efficiency of digital images, and lower the network cost. JPEG images are frequently stored and transmitted over the Internet and are utilized in various devices, such as digital cameras and scanners. Designing an encryption and compression technique with exceptional reconstruction ability is crucial, because it is not practical to directly compress the data in the encrypted region due to the inability of existing compressors to handle the encrypted data structure. Furthermore, the compressed image encryption technique is challenging, because it requires increasing the image's entropy to boost security while lowering the image's bit rate to improve compression efficiency. Johnson et al. showed that the security and compression effectiveness of binary encrypted images remain unaffected by the application of coding algorithms based on the idea of auxiliary information [15]. By altering the source memory, Schonberg et al. increased the compression efficiency of encrypted images, which has the benefits of high encryption source secrecy and fewer bits needed [16]. Block scrambling image encryption is used in encryption-then-compression (EtC) techniques to guarantee that EtC images are unaffected by interpolation, leading to better compression performance than conventional

approaches [4, 8–11]. Additionally, because the secret key space of EtC images is sufficiently large, they are highly resistant to brute-force attacks.

For requirement 3, one efficient method for users to identify the desired image effectively is to encrypt only its secret region. In addition to reducing the processing time, partial encryption enables the encryption of particular regions in accordance with each individual's needs [17]. In addition, users do not need any further information or technologies to identify the encrypted image on the public network by obtaining the nonsecret region [18]. Wen et al. used optical encryption and chaos on the image's prominent areas to create visually meaningful encrypted images [19]. Khan et al. employed a lightweight image encryption technique that computes the correlation coefficient and blocks the original image to improve security [20]. The plaintext region is visible to the human eye, while the secret region is rendered unintelligible and unbreakable by the encryption method. However, data hiding is an additional technique for image protection that may be used to conceal information in unencrypted formats, such as text, audio, video, and images. Since only the sender and the recipient are aware that the information is embedded in the sent image, private information can be safely transferred across a public network by making it invisible to an attacker. Ju et al. conducted an experiment to demonstrate how the differential privacy combination theorem-based watermarking technique can successfully lower the privacy budget [21]. This research presents a region-selective anti-compression image encryption algorithm based on deep networks that combines the benefits of the previously mentioned techniques. We first employ the Peano mask tree to represent the position information of the secret regions for the original image. To obtain the encrypted image, we encrypt the secret regions during the encryption phase using a block scrambling approach based on a chaotic system. Finally, we embed the secret information into the encrypted image using an upgraded deep learning-based embedding network to obtain the embedded image. Our main contributions are as follows:

- We provide a novel image encryption scheme that simultaneously meets three requirements: the security of image privacy can be adequately guaranteed, and after JPEG compression, the image can be successfully decrypted; moreover, this approach is visually meaningful due to the region selection technique.
- The proposed scheme has excellent robustness against JPEG compression attacks because of the improved data-hiding techniques based on deep learning.
- Experiments prove that the proposed scheme is effective and has good application prospects for social network services.

The rest of the paper is organized as follows. In Sect. 2, we review the cited related work. In Sect. 3, we introduce the details of the proposed scheme in detail and present the experimental results and discussion in Sect. 4. Finally, Sect. 5 concludes.

2 Related Work

Different approaches, such as data hiding and image encryption, have been applied to image technology to guarantee the security of image privacy. These techniques have shown their benefits in various uses [22]. Traditional cryptography, such as symmetric key cryptography, offers no advantage in regard to image encryption for images used in network services, since the algorithm complexity is too high and more encryption rounds are needed. In contrast, image encryption using lightweight techniques such as image blurring, mosaic masking, deletion, and distortion is more common. These techniques nearly always involve two stages: diffusion and confusion. Blurring makes the image difficult to recognize, while mosaic masking weights the image through the tile form. Deletion and distortion remove data from parts of an image and randomly modify image pixel values. Using two-point crossover and uniform mutation algorithms in genetic algorithms, Gupta et al. devised a lightweight symmetric image encryption technique [23]. This algorithm's lightweight design and quick algorithmic speed make it appropriate for use in Internet-of-Things devices. A color image encryption technique based on block scrambling and chaotic systems was utilized by Hosny et al. [24]. The original image was divided into three identical channels, each of which was further divided into diffuse and confused image blocks of varying sizes. There are two types of chaotic systems: low-dimensional chaotic maps and high-dimensional chaotic maps. While low-dimensional chaotic maps have low temporal complexity and good efficiency, high-dimensional chaotic maps have excellent security. The proposed scheme employs a chaotic system-based block scrambling image encryption technique, which renders it vulnerable to changes in the secret key. At the same time, the encrypted image can withstand JPEG compression, because the proposed approach works with pixel blocks rather than individual pixels.

Whole image encryption for images uploaded to SNSs frequently results in more repetitive tasks, since users are primarily concerned with the security of their personal information, such as an image's license plate or headshot. Therefore, it is better in accordance with the user's need to encrypt only the secret region when encrypting the entire image. Reversible data hiding (RDH) is used for partial encryption to pinpoint the secret regions of the original image. Because the typical secret region is rather large,

embedding the unprocessed secret region into the image may result in a significant waste of computer processing time and increased expense. Wang et al. used the PSPNet semantic segmentation model to extract images' relevant portions (for example, people) and then used an encryption algorithm based on a chaotic system to encrypt irregularly sized pixel blocks [25]. Ping et al. developed a resilient image encryption algorithm based on life-like cellular automata to address the challenge of encrypting regions of interest of diverse sizes and types in medical images [26]. Based on the pixel continuity of the hidden sections, Qin et al. integrated the text data into the image by compressing it using the Peano mask tree (PM-tree) [27]. The proposed scheme employs the PM-tree to treat the encrypted position as secret data and the pixel scrambling algorithm to encrypt the secret region to enable the algorithm to handle irregular regions and lower computational costs.

In some sensitive circumstances, such as military and legal forensics, data hiding is frequently employed to recover the carrier medium and recover the hidden information simultaneously without losing any content. Digital watermarking and steganography are the two directions that make up the data-hiding algorithm. The embedding algorithm and the extraction method are the two crucial parts of both methods. Steganography's goal is to shield confidential information from being discovered by unapproved instruments; the image that is produced is known as a stego image. An image created by digital watermarking is known as a watermark image, and its goal is to safeguard the confidentiality of the secret data. Luo et al. used a stereo matching-based image watermarking approach for the web service authentication of stereo images [28]. However, image manipulation algorithms, such as noise, rotation, and compression, can readily damage traditional data-hiding techniques, destroying the embedded information. The classic matrix embedding approach was tested in terms of security when Chen et al. devised a differential attack algorithm based on matrix embedding [29]. Because machine learning tools can offer more effective methods for embedding and extracting secret data, their use in data-hiding strategies is becoming increasingly common [30]. Singular value decomposition is the foundation of Yang et al.'s neural network watermarking algorithm, which qualifies the algorithm for audio copyright protection [31]. Sinhal et al. presented a novel deep neural network-based image watermarking method [32]. This method can determine the source of digital content shared or transferred across the Internet at various social networking sites. Recurrent convolutional layers are used in Ahmadi et al.'s residual-based diffusion watermarking system, which exhibits good robustness against a variety of attacks, such as Gaussian, compression effects, and cropping attacks [33]. The proposed scheme strengthens the robustness of the embedded image to JPEG compression using an enhanced residual diffusion-

Table 1 Summary of the literature survey

Algorithm	Methods/techniques	Low complexity	Anti-compression	User-friendly	Color support	Capacity
Chuman et al. [9]	Block scrambling	×	✓	×	×	×
Wang et al. [25]	Low-dimensional chaotic system	✓	×	✓	✓	×
Anbarjafari et al. [30]	Z-transform and QR decomposition	✓	✓	×	✓	✓
Kenta et al. [34]	Block scrambling	×	✓	×	✓	×
khan et al. [20]	Chaotic maps	✓	×	×	✓	×
Ahmadi et al. [33]	CNN	✓	✓	×	×	✓
Gupta et al. [23]	Crossover and mutation operators	×	×	×	✓	×
kurihara et al. [11]	Block scrambling	×	✓	×	✓	×
Huang et al. [35]	Least significant bit method	✓	×	×	×	✓
Proposed	CNN	✓	✓	✓	×	✓

based embedding algorithm. Table 1 provides a summary of the literature review.

3 Compression-Resistant Image Encryption Model

We propose a deep learning-based image encryption model to preserve privacy. Rather than encrypting the entire image, our model is a region-based approach, allowing users to control the encryption of specific regions of the image. The proposed design consists of several stages: (a) region indicator stream generation, (b) secret content encryption, (c) information embedding and extraction, and (d) network training.

To handle different numbers and sizes of regions of secret (RoS), we employ a Peano mask tree to represent and compress the RoS position information. The corresponding bit stream of RoS position information, named the region indicator stream (RIS), is then utilized for further information embedding. Additionally, to enhance resilience against plaintext crackers, the image region of RoS undergoes encryption using a block-based scrambling method. We utilize a deep learning-based model for our information embedding system, enabling the hiding of the RIS within an image. This facilitates the image's ability to withstand compression operations with lower quality factors. It is widely recognized that learning-based methods can be readily deployed in domain-specific applications by providing appropriate training samples, and the system's performance also highly relies on the quantity of the available samples. With regard to web applications, we emphasize the robustness of Image JPEG recompression as a training requirement while extracting embedded information during network training. The entire system framework is illustrated in Fig. 1, and the primary notations and descriptions are listed in Table 2.

3.1 Region Indicator Stream Generation

In the proposed method, it is essential to accurately distinguish between RoSs and non-RoSs when encrypting an image. The RoS position information is effectively embedded into the encrypted image for image decryption and reconstruction. However, directly embedding RoS information without proper representation can result in unnecessary space occupation and poor embedding performance. Considering that the pixels within RoS are contiguous, we can leverage the RoS position information as a region indicator stream and adopt the Peano mask tree (PM-tree) to represent the RoS positions. The PM-tree is a quadrant-based lossless compression method that enables efficient calculation of the original spatial data while fully retaining the data information.

RoS position map For a given image $I \in \mathbb{R}^{W \times H}$, W and H are the width and height of the image, respectively. The secret regions are marked as RoSs; otherwise, as non-RoSs, both are given by the user. To store the position of RoSs, a matrix M , a position map, $M = [m_{i,j}] \in \{0, 1\}^{W \times H}$, having the same size as the image I , is used to store the RoSs position, where $m_{i,j} = 1$ is labeled 1 with the pixels in RoSs and $m_{i,j} = 0$ is labeled 0 for the non-RoSs.

Peano Mask Tree The Peano Count Tree (P-tree) is a quadrant-based lossless tree representation of spatial data. The concept of the P-tree is to recursively divide the map M into hierarchical quadrants $M_{l_1 \dots l_k \dots} = [m_{s,t}^{l_1 \dots l_k \dots}] \in \{0, 1\}^{W_k \times H_k}$, $W_k = \frac{W}{2^k}$, $H_k = \frac{H}{2^k}$, $l_k = 0, 1, 2, 3$, $k = 1, \dots, K - 1$ and $K = \lceil \log_4(W \times H) \rceil$ and record the counts of '1' and $C_{l_1 \dots l_k \dots} = \sum_{s,t} m_{s,t}^{l_1 \dots l_k \dots}$ in each quadrant, thus forming a quadrant count tree, where $m_{i,j}^{l_1 \dots l_{k-1}} = m_{s,t}^{l_1 \dots l_{k-1} l_k}$ with $i = s + \mathbb{1}_{[l_k=1 \vee l_k=3]} W_k$ and $j = t + \mathbb{1}_{[l_k=2 \vee l_k=3]} H_k$. For example, the map M corresponding to the root node with value $C = \sum_{i,j} m_{i,j}$ is divided into four equal quadrants, $M_{l_1} = [m_{s,t}^{l_1}] \in \{0, 1\}^{\frac{W}{2} \times \frac{H}{2}}$, $l_1 = 0, 1, 2, 3$,

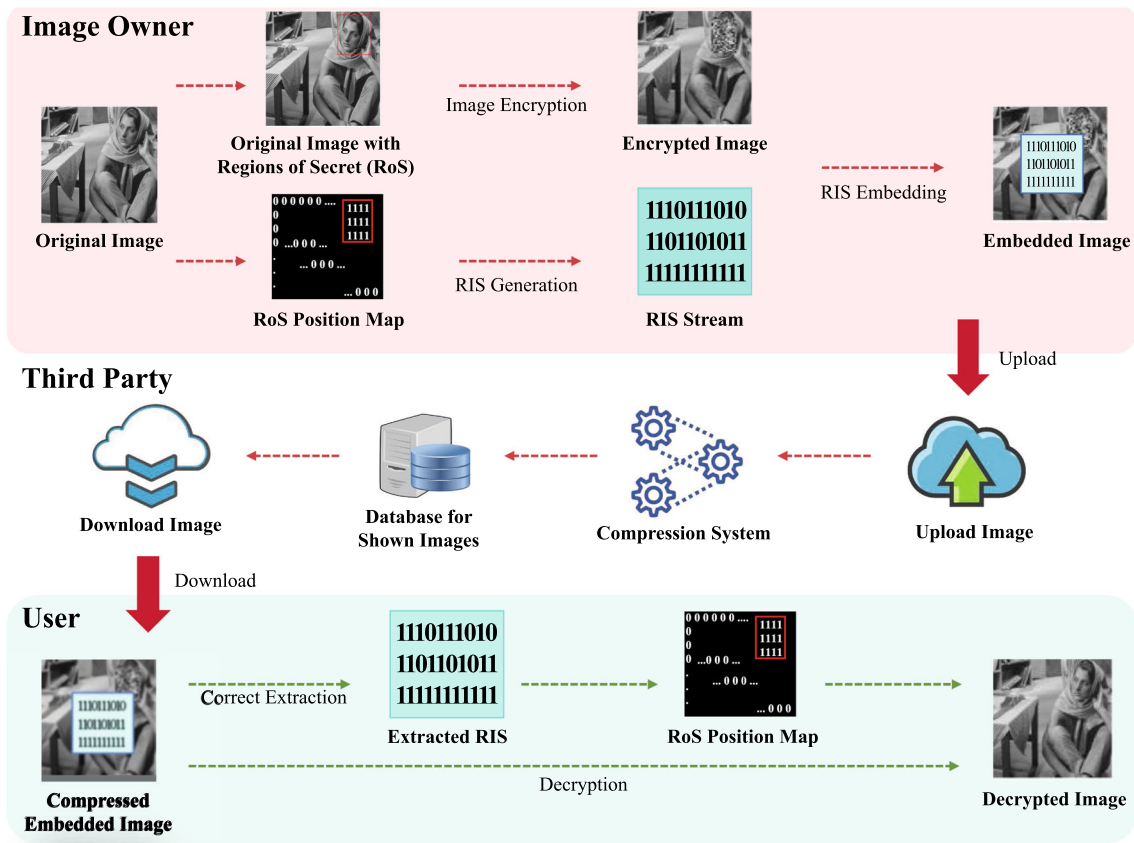


Fig. 1 The framework of the proposed encryption scheme

Table 2 Primary notations and descriptions

Notations	Descriptions	Notations	Descriptions
I	Original image	EXT_{ϕ}	Extraction network
\tilde{J}	Rearranged image	$\tilde{P}^{(k)}$	Depth-based vectorized data
J	Encrypted image	$\tilde{\mathfrak{P}}^{(k)}$	Output of the DCT transformation
E	The embedded image	r_s	Nonoverlapping and disjointed secret region
$m_{i,j}$	RoS mark for each pixel	\mathcal{I}_s	Corresponding pixel set coordinates
B	Bitstream of RIS	L	Length of pseudorandom numbers
M	Position map to store the position of RoSs	p_s	Number of blocks divided by r_s
\mathcal{J}	Indices of V for the elements in O	H_s	The scrambling of the pixels I_s
$\Theta(\cdot, \cdot)$	Rotation and horizontal and vertical inversion transformation	W	Width of original image
$P^{(k)}$	Nonoverlapping rectangular image parts	α	Strength factor
ξ/ρ	Control state parameters for chaos dynamics	H	Height of original image
δ	Discarded pseudorandom numbers on the sequences	γ	Weight to control imperceptibility and robustness

and the value of $C_{l_1} = \sum_{s,t} m_{s,t}^{l_1}$ is the count of ‘1 s’ in the M_{l_1} quadrant.

When the quadrant is composed entirely of ‘1 s’, that is, if the number of ‘1 s’ is equal to the size of the quadrant, no subtree is needed; it is a leaf node. Another case of a leaf node is a quadrant composed entirely of ‘0’. The PM-tree is a further compressed structure tree of the P-tree that uses a

mask instead of a count. If a node is a leaf node, the value of the PM-tree is 0 if the count is 0; otherwise, it is 1. If it is a nonleaf node, the value of the PM-tree is masked to ‘M’. An example of a PM-tree is shown in Fig. 2.

Region Indicator Stream Finally, the PM-tree of an RoS position map is compressed into a bitstream. The node values in the PM-tree are further replaced by ‘11’, ‘00’ and ‘10’ for

‘M’, ‘0’ and ‘1’, respectively, and then, a preorder traversal is used to record the PM-tree to generate the corresponding bitstream. For the embedding requirement, the bitstream will be padded with code ‘11’ at the end to a predefined length as the RIS. According to the definition of code ‘11’ for the mark ‘M’ in the PM-tree, which represents a nonleaf node, padding ‘11’ to the end of the bitstream is reasonable and does not cause harm, as shown in Fig. 2d. The bitstream of the RIS consists of binary data, $B = [b_k] \in \{0, 1\}^K$, which are embedded in the image for image decryption and reconstruction, as discussed in Sect. 3.3.

3.2 Secret Content Encryption

We consider an EtC image application, which refers to a lossy image compression system in the cryptographic domain. The images are first encrypted by the owner, then compressed by the provided channel, and finally decompressed and decrypted by the receiver. We utilize a block-based scrambling encryption scheme to encrypt the region of the RoSs in the original image. This scheme not only demonstrates compression performance comparable to that of the original scheme for encrypted JPEG images but also exhibits greater security and robustness. The proposed RoS block-based scrambling encryption scheme comprises four main steps: pixel-based scrambling, block-based scrambling, block rotation and inversion, negative-positive transformation, and block reconstruction. These steps are illustrated in Fig. 3.

For a given RoS position map M and an image I , each nonoverlapping and disjoint sensitive region RoS r_s , $s = 1, 2, \dots, S$, can be an irregular shape, the acquired pixels of RoS r_s are denoted by $I_s = I(r_s) = \{I_s^{(k)}\}_{k=1}^{t_s}$, and the corresponding pixel coordinates $\mathcal{I}_s = \mathcal{I}(r_s) = \{(i_s^{(k)}, j_s^{(k)})\}_{k=1}^{t_s}$, where t_s is the number of pixels in r_s , S is the number of nonoverlapping and disjoint RoS, i.e., $\mathcal{I}_s \cap \mathcal{I}_t = \emptyset$, $s \neq t$, and $|\bigcup_s \mathcal{I}_s| = \sum t_s$, both elements in I_s and \mathcal{I}_s are ordered coordinates lexicographically, i.e., $k \leq k'$ if and only if $i_s^{(k)} < i_s^{(k')}$ or $(i_s^{(k)} = i_s^{(k')})$ and $j_s^{(k)} \leq j_s^{(k')}$. Therefore, to perform block-based processing, we can integrate all of the pixels in I_s as a one-dimensional array, convert I_s into a $w_s \times h_s$ matrix, and then perform block-based scrambling encryption. Following encryption, these RoS pixels are returned to the original image region using the same guidelines; to facilitate discussion, let w_s, h_s be the closest common divisor pair of t_s . This process is illustrated in Fig. 3a.

- **Random number generation** Chaos-based encryption models were found to be fast and efficient. To achieve high security in image encryption, we adopt the LS chaos encryption model in both pixel-based and block-based manners. The adopted LS model embedded a sine map into a logistic map and was found to be highly random

and sensitive to the control parameters and initial conditions. For pixel set $I_s = \{I_s^{(k)}\}_{k=1}^{t_s}$ of the RoS r_s , we first generate L unpredictable unique pseudorandom numbers $u_i \in (0, 1)$, $i = 1, 2, \dots$ based on the LS chaotic dynamics equation

$$u_{i+1} = (\xi + (4 - \xi)u_i) \times \left| \frac{\rho}{4} \sin(\pi u_{i-1}) \right| \times \left(1 - \left| \frac{\rho}{4} \sin(\pi u_{i-1}) \right| \right), \quad (1)$$

where $u_1, u_2 \in (0, 1)$ are initial values, $\xi \in [3, 4]$ and $\rho \in [1.4, 4]$ are control state parameters for chaos dynamics, $L = t_s + 3 \times p_s + \delta$, t_s is the number of pixels of the RoS r_s , p_s is the number of blocks divided by r_s , which will be discussed later, and δ , the first δ , discard pseudorandom numbers on the sequences. Here, the user control key of RoS is $(u_1, u_2, \rho, \xi, \delta)$.

- **Pixel Scrambling** We scramble the pixels in $I_s = \{I_s^{(k)}\}_{k=1}^{t_s}$ based on reordering. We take t_s random numbers from the LS model $V = \{v_k | v_k = u_{\delta+k}, k = 1, 2, \dots, t_s\}$ and rearrange V in descending order to obtain two lists, $O = \{o_k\}_{k=1}^{t_s}$, ($o_1 > o_2 > \dots > o_{t_s}$), the result of rearranging V , and $\mathcal{J} = \{j_k\}_{k=1}^{t_s}$, the indices of V for the elements in O

$$\begin{aligned} O &= \{o_k | o_k \in V \text{ s.t. } \#\{v \in V | v \geq o_k\} = k\} \\ \mathcal{J} &= \{j_k | j_k \in [t_s] \text{ s.t. } j_k = \min\{i | o_k = v_i\}\}. \end{aligned} \quad (2)$$

The scrambling of the pixels I_s is obtained by

$$H_s = \{h_s^{(k)} | h_s^{(k)} = I_s^{(j_k)}, k = 1, 2, \dots, t_s\}. \quad (3)$$

- **Block scrambling** As illustrated in Fig. 3a, we convert the RoS pixel array H_s from a one-dimensional array to a $w_s \times h_s$ matrix. As shown in Fig. 3b, block scrambling involves dividing the matrix version of H_s into p_s nonoverlapping block-based pixel groups, where $B_s = \{b_s^{(k)}\}_{k=1}^{p_s}$, and randomly permuting the order of B_s , where $p_s = \lfloor \frac{w_s}{8} \rfloor \times \lfloor \frac{h_s}{8} \rfloor$. The process for scrambling blocks is the same as that for pixels; we pick p_s random numbers from the LS model, $V' = \{v'_k | v'_k = u_{\delta+t_s+k}, k = 1, 2, \dots, p_s\}$, and reorder V' to decrease the order of the blocks B_s .
- **Block rotation and inversion** Random block inversion and rotation are performed next to the encryption process, followed by block scrambling. As Fig. 3b shows, *block rotation* represents a random rotation of pixel blocks by $0^\circ, 90^\circ, 180^\circ$, or 270° , and *block inversion* represents the horizontal and vertical inversion of pixel blocks. Based on the random number obtained by the LS model, we apply a random rotation and a horizontal and vertical inversion on $b_s^{(k)}$ for each block, $b_s^{(k)} \in B_s$. The LS model's

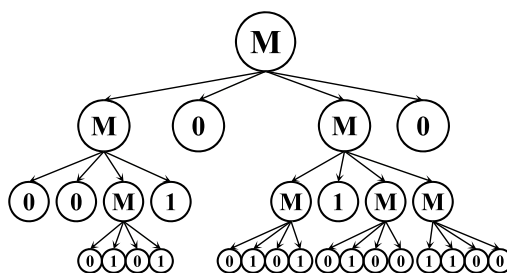
Fig. 2 An example of a region indicator stream



(a) Example of original image with RoS

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0
0	1	1	1	0	0	0	0
0	1	1	1	0	0	0	0
0	1	1	1	0	0	0	0
0	1	1	1	0	0	0	0
0	0	0	0	0	0	0	0

(b) Example of encrypted position



(c) PM-Tree



```
11110000110000111011
10111100100010001000
100010000010100000..
...111
```

(d) Region indicator stream (RIS)

p_s random numbers, $V'' = \{v_k'' | v_k'' = u_{\delta+t_s+p_s+k}, k = 1, 2, \dots, p_s\}$, are used to rotate and reverse the block $b_s^{(k)}$

$$r_k = 1 + \left(\left\lfloor |v_k''| \times 10^{12} \right\rfloor \bmod 16 \right), \tag{4}$$

$$\tilde{b}_s^{(k)} = \Theta(b_s^{(k)}, r_k),$$

where $\Theta(\cdot, \cdot)$ is a rotation and horizontal and vertical inversion transformation.

- **Negative-positive transformation** Random negative-positive pixel transformation is the next phase in the encryption process. This phase involves calculating the pixel values in each block $\tilde{b}_s^{(k)}$ using the random number produced by the LS model. From the LS model, $V''' = \{v_k''' | v_k''' = u_{\delta+t_s+2 \times p_s+k}, k = 1, 2, \dots, p_s\}$, we select p_s random numbers. The pixel value $\tilde{I}_{\tilde{b}_s^{(k)}}(x, y)$ at the (x, y) position of $\tilde{b}_s^{(k)}$ is determined by

$$\tilde{I}_{\tilde{b}_s^{(k)}}(x, y) = \begin{cases} I_{\tilde{b}_s^{(k)}}(x, y), & r'_k = 0 \\ 255 - I_{\tilde{b}_s^{(k)}}(x, y), & \text{otherwise,} \end{cases} \tag{5}$$

where r'_k is a random binary number generated based on the LS model as

$$r'_k = 1 + \left(\left\lfloor |v_k'''| \times 10^{14} \right\rfloor \bmod 2 \right). \tag{6}$$

- **Region Reconstruction** The processed matrix needs to be converted back into a one-dimensional pixel array after the previously outlined procedures. We can fill the one-dimensional pixel array back into the original image using the RoS's position data. We successfully encrypt the hidden areas of the image I to an encrypted image J by finishing this procedure.

3.3 Information Embedding and Extraction

An information flow block diagram with three primary modules: a CNN for embedding the RIS, a microattack layer for mimicking JPEG attacks, and a CNN for retrieving buried RIS is shown in Fig. 4. The structure of the embedding network consists of two nontrainable transform layers, a discrete cosine transform (DCT) layer, inverse DCT transform layers, and seven trainable convolution-based layers. The embedding network, $E = \text{EMB}_{\theta}(J, B; \alpha)$, has three inputs, an encrypted image $J \in \mathbb{R}^{W \times H}$, K -bit binary data $B = [b_k] \in \{0, 1\}^K$, and a strength factor $\alpha \in \mathbb{R}_+$, where θ denotes the network parameters. The details of the pipeline are discussed below and shown in Fig. 4.

Embedding Layer For embedding K bits, we assume that K can be factorized as $K = S \times R$. The encrypted images $J \in \mathbb{R}^{W \times H}$ are divided into K nonoverlapping rectangular

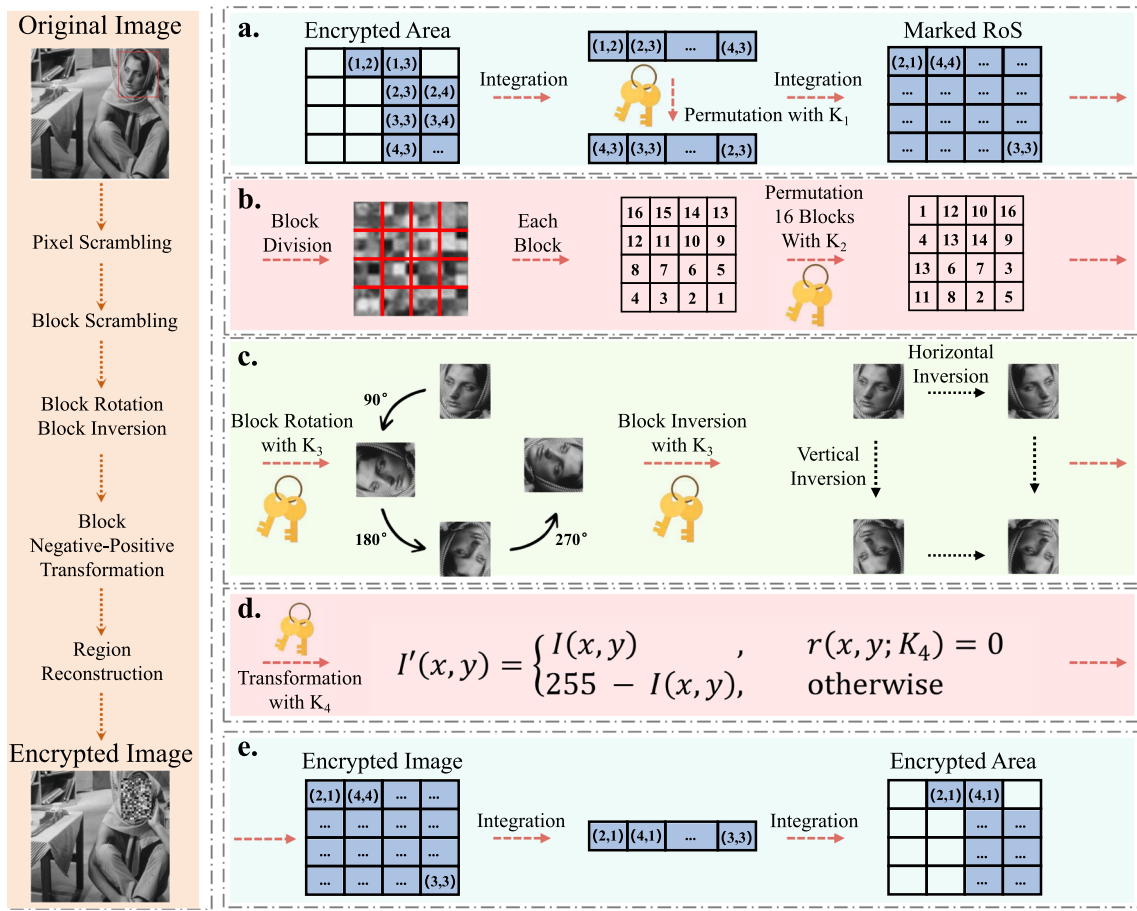


Fig. 3 The framework of the proposed encryption scheme

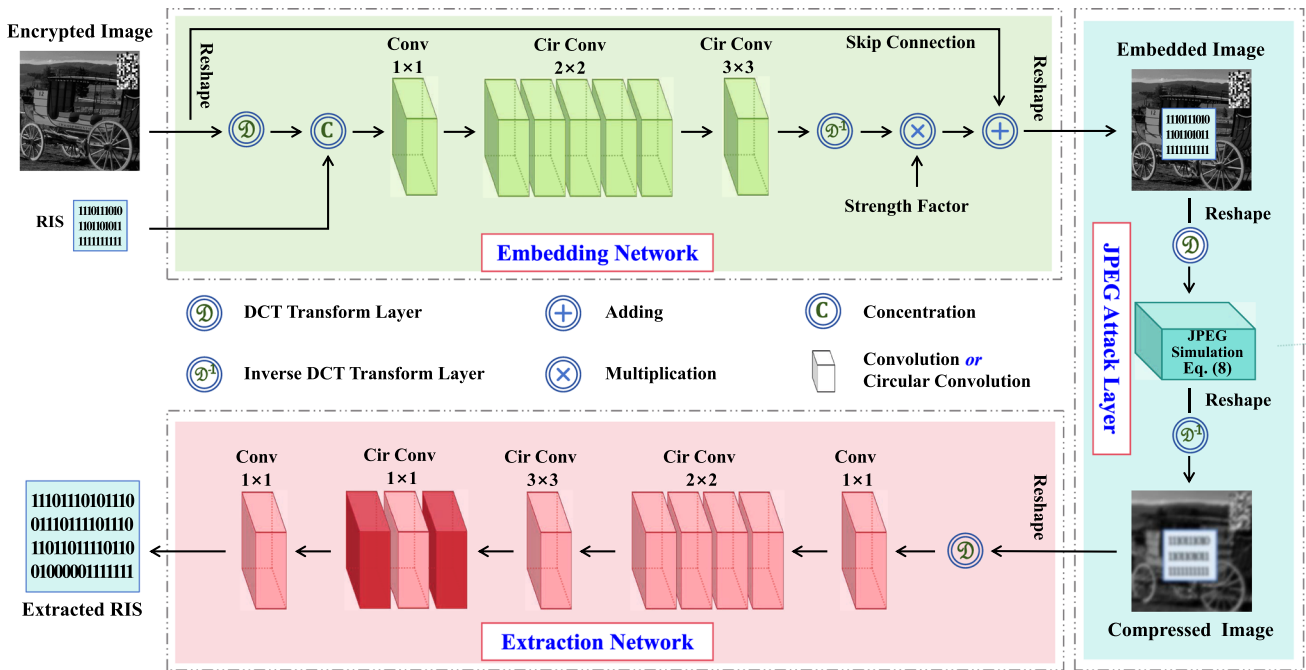


Fig. 4 The framework of the proposed embedding scheme

image parts, $P^{(k)} = [p_{s,r}^{(k)}] \in \mathbb{R}^{M \times N}$, $k = 1, 2, \dots, K$, $M = \frac{W}{S}$, and $N = \frac{H}{R}$, and each image part $P^{(k)}$ is used to store one bit of information b_k . To spare the bit information b_k to all the pixels of the image part $P^{(k)}$, a space-to-depth operation is applied to rearrange spatial pixel data $P^{(k)} \in \mathbb{R}^{M \times N}$ into depth-based vectorized data $\tilde{P}^{(k)} = [\tilde{p}_{1,1,d}^{(k)}] \in \mathbb{R}^{1 \times 1 \times MN}$ and will work together with convolution operations. After a space-to-depth operation, the rearranged image \tilde{J} of J is a tensor $\tilde{J} \in \mathbb{R}^{S \times R \times MN}$, the mapping of elements from $J_{i,j}$ to $\tilde{J}_{s,r,d}$ by $s = \lfloor i/M \rfloor$, $r = \lfloor j/N \rfloor$ and $d = (j - rN) \times M + (i - sM)$.

To embed B to \tilde{J} , we further reshape and concatenate $B \in \{0, 1\}^{S \times R}$ with $\tilde{J} \in \mathbb{R}^{S \times R \times MN}$ in depth as $[\tilde{J}, B]$, and the resulting concatenation is a tensor of size $S \times R \times (MN + 1)$. One 1×1 normal convolutional layer, five 2×2 circular convolutional layers and one 3×3 circular convolutional layer are composed sequentially as embedding layers. The last 3×3 layer spreads the information over a wider neighborhood, making the embedded images more secure against JPEG attacks.

The embedding layers are expected to learn to embed and distribute the binary data among adjacent image parts as well as the depths. To obtain the final embedded image E with the same size of input J , at least the number of filters of the last final convolution layer must be MN . As a result, the output of the embedding layers is a tensor of size $S \times R \times MN$; then, a depth-to-space rearrangement operation will reshape the tensor back to an image E , which is the same size as J .

Transform Layer The JPEG compression algorithm works on the DCT domain, and it is suggested that the embedding network computes the binary data embedding in the DCT domain, as shown in Fig. 4. The DCT/inverse DCT transform layer implements a reversible linear transform that changes the representation of an image from the spatial domain to the DCT domain and vice versa. We place a DCT transform and an inverse DCT transform layer before and after the embedding layers, respectively. Considering DCT compression as a block-based compression, with the rearrangement of images \tilde{J} discussed in the previous section, the DCT transformation layer is simplified to a 1×1 convolution layer and applied to each image part $P^{(k)}$ independently. The output of the DCT transformation $\tilde{\mathfrak{P}}^{(k)} = [\tilde{p}_{1,1,\tau}^{(k)}] \in \mathbb{R}^{1 \times 1 \times MN}$ of image part $\tilde{P}^{(k)} \in \mathbb{R}^{1 \times 1 \times MN}$ at depth τ is calculated by

$$\begin{aligned} \tilde{\mathfrak{P}}_{1,1,\tau}^{(k)} &= \frac{1}{MN} \sum_{s=0}^{M-1} \sum_{r=0}^{N-1} p_{s,r} \cdot \underbrace{\cos\left(\frac{(2s+1)u\pi}{2M}\right) \cos\left(\frac{(2r+1)v\pi}{2M}\right)}_{g(s,r;u,v)} \Bigg|_{\substack{s=\lfloor \frac{\eta}{M} \rfloor \\ r=\eta-sM \\ u=\lfloor \frac{\tau}{M} \rfloor \\ v=\tau-uM}} \\ &= \sum_{\eta=0}^{MN-1} \tilde{p}_{1,1,\eta}^{(k)} \cdot \left[(MN)^{-1} \cdot \tilde{g}(\eta, \tau) \right] \\ &= \langle \tilde{P}^{(k)}, \tilde{G}(\tau) \rangle, \end{aligned} \tag{7}$$

where $g(s, r; u, v)$ is a two-dimensional DCT transformation kernel and $\tilde{g}(\eta, \tau)$ is g after changing the variables. \tilde{G} can be viewed as a convolutional layer with $\tilde{G}(\tau)$ as the τ -th convolution mask. For the implementation, $\tilde{G}(\tau)$ is a 1×1 convolution mask with MN input channels representing the weights of one neuron with no bias, which is only a fixed weight convolutional layer with MN convolution masks.

Embedding and Extraction Network The final structure of the embedding network is shown in Fig. 4, which is a residual-based structure. Skip connections have been shown to be efficient in network training. The final embedded image E is the sum of the outputs of the embedding layers and the original encrypted image. Instead of simply summing the results of the embedding layers with the original input, a strength factor $\alpha \in \mathbb{R}_+$ obtained via multiplication is used to adjust the strength of the embedding output.

For the extraction network, $H = \text{EXT}_{\phi}(E)$ has an input of a binary data embedded image $E \in \mathbb{R}^{W \times H}$, and an output of $H \in [0, 1]^K$ is the probability of the extracted k -bit binary data, where ϕ denotes the network parameters, as shown in Fig. 4, which is supposed to extract the embedded binary data from the input image. One notable difference is that the structure of the extraction network differs from that of the embedding network. Several modifications are made while preserving the five convolutional layers and one DCT transformation layer to enhance the performance of the extraction network. Specifically, three additional layers of 1×1 circular convolutions are introduced to improve information extraction. Among these layers, two additional convolutional layers (shown in red Fig. 4) are expanded by a factor of 2 while reducing the number of channels by half. This adjustment aims to optimize the dimensions of the network's output, ultimately enhancing the extraction performance. By incorporating these modifications into the extraction network, improved results can be achieved in extracting the hidden information stream.

3.4 Network Training

The trainable network combines an embedding network, a JPEG simulation layer, and an extraction network to form an end-to-end trainable network, as shown in Fig. 4.

The JPEG simulation layer simulates JPEG compression–decompression, as a differentiable network layer can facilitate the backpropagation of the training gradient and allow the embedding and extraction networks to train more robust embedded images during the training phase for JPEG compression.

JPEG Simulation Layer There are several nondifferentiable operations involved in the JPEG compression–decompression process. Thus, we need a differentiable approximation to simulate the process in a network layer. The steps of JPEG compression are as follows: a) divide the image into a nonoverlapping 8×8 block and computer block-based DCT transformation; b) quantize these DCT coefficients by dividing by a predefined quantization matrix Q and rounding the result to the nearest integer, which is a lossy and nondifferentiable step; and c) apply a lossless compression algorithm. Similarly, JPEG decompression involves inverse operations, d) decompression, e) dequantization of the coefficients by multiplying the same quantization matrix Q , and f) transformation to the spatial domain by inverse DCT.

In the proposed network, the JPEG simulation layer simulates JPEG compression and decompression via the following steps: a) DCT transform, b) quantization, e) dequantization, and f) inverse DCT transform. All the rounding operations in b) are nondifferentiable, and an approximation is needed to facilitate backpropagation of the training gradients. The rounding operation is simulated by adding a uniform noise in the range $[-0.5, 0, 5]$ to the coefficients as

$$\tilde{p}_{1,1,\tau}^{*(k)} = \left(\frac{\tilde{p}_{1,1,\tau}^{(k)}}{Q} + \beta \right) \times Q = \tilde{p}_{1,1,\tau}^{(k)} + \beta, \quad (8)$$

where $\beta \sim U(-0.5, 0.5)$. Then, the overall process simulates a JPEG compression–decompression of an image.

Training Objective The embedding network and the extraction network have independent objective functions. The embedding network is supposed to generate an embedded image with maximum quality and minimum distortion, and the extraction network is responsible for maximizing the extraction rate of the hidden binary data. For the embedding network, we use the structural similarity index (SSIM) as a loss function to measure the degradation of image quality due to position information embedding

$$L_1 = \text{SSIM}(J, E) = \frac{(2\mu_J\mu_E + c_1)(2\sigma_{J,E} + c_2)}{(\mu_J^2\mu_E^2 + c_1)(\sigma_J^2\sigma_E^2 + c_2)}, \quad (9)$$

where J is the input image of the embedding network, E is the output of the embedding network, μ_J and μ_E are the mean values of J and E , respectively, σ_J and σ_E are their variances, and $\sigma_{J,E}$ is the covariance of (J, E) . During training, we set the constants c_1 and c_2 to 10^{-4} and 9×10^{-4} , respectively.

For the extraction network, we use binary cross-entropy loss

$$\begin{aligned} L_2 &= \text{CE}(B, H) \\ &= - \sum_{i=1}^K b_i \log(h_i) + (1 - b_i) \log(1 - h_i), \end{aligned} \quad (10)$$

, where $B = [b_i] \in \{0, 1\}^K$ is the original binary data and $H = [h_i] \in [0, 1]^K$ is the output of the extraction network, which is the probability of the k binary bits B , $\mathbb{P}\{H = B\}$. The total loss is a weighted sum of the two losses, $L_{\text{total}} = \gamma L_1 + (1 - \gamma)L_2$, where $\gamma \in [0, 1]$ is the weight to control imperceptibility and robustness.

In summary, the training network is an end-to-end trainable network composed of $E = \text{EMB}_{\theta}(J, B)$, $E' = \text{att-JPEG}(E)$, and $H = \text{EXT}_{\phi}(E')$, and the objective function $\min_{\theta, \phi} L$ is minimized for network training.

4 Experimental Results

4.1 Experimental Setup and Evaluation Metrics

TensorFlow [36] implements the proposed training network during the training phase, which is run on a GeForce GTX 1070. In the experiments, we use 32×32 grayscale images as the training input and fix the block size to 8×8 , so each input image includes 16 blocks. Simultaneously, we employ 4×4 random binary bits as the RIS to avoid utilizing a fixed embedding input, which could cause bias in the network. During training, there are 64 convolutional filters in all layers of the extraction network and embedding network, except for the extraction network's labeled convolutional filters, which have 128 and 32 filters, respectively. The filters' strides are also 1. The sigmoid-activated neurons in the extraction network are positioned in the final 1×1 convolutional layers to generate the information probability map, which is then processed using the threshold operation to produce the secret information. For training set selection, we combined the Pascal VOC2012 [37] and CIFAR10 [38] datasets to form the patch set to train the network. The CIFAR10 dataset comprises 60,000 RGB images of size 32×32 , of which 50,000 are used for training and 10,000 for testing. We take random 32×32 pixel blocks from the large high-resolution images in the Pascal dataset. The patch set includes more than 333k grayscale image blocks after the chosen pixel blocks have undergone grayscale processing. The training procedure has 1 million iterations and takes approximately 6 h. Table 3 contains the training parameters. After the training phase, the embedding and extraction layers are used as separate networks.

During the simulation stage, we retrieve the hidden data and use the real JPEG recompression to evaluate the robust-

Table 3 The training parameters of the data-hiding framework

Parameter	Value	Description
(W,H)	(32,32)	Size of training image
(M,N)	(8,8)	Size of divided image part $P_{s,r}$
(S,R)	(4,4)	Size of reshaped binary data B
Epoch	100	Training epoch number
LR	10^{-4}	Learning rate
λ	0.75	Relative loss function weights

ness. We use the Fdez-Vidal dataset [39], which contains 49 grayscale images as test images, with each test image having a size of 512×512 , to assess the proposed approach. During the encryption region selection stage, we artificially designate sensitive positions in the image, such as people, buildings, and cars. During the RIS-generating phase, the acquired encrypted position information is processed to produce the RIS, which is typically limited to $K_{\text{limit}} = 1024$ bits or 32^2 . For the encryption phase, we set $U_1=0.4267$, $U_2=0.4247$, $\rho = 3.96$, $\xi = 3.9$ and $\delta = 1000$. The generated encrypted image and RIS are passed to the embedding network as input. When comparing the training parameters (Table 3), the trained network has an embedding capacity of $c = 4^2/32^2$ bits per pixel (BPP). Given an embedding capacity c for a 512×512 image, we may embed $K = 4^2/32^2 \times (512 \times 512) = 4,096$ bits. This capacity is four times greater than our requested limit K_{limit} . As a result, the RIS contained in the input pieces has four times more redundancy, which greatly improves its security.

Multiple JPEG compression attacks were applied to the resulting embedded images to mimic real-world scenarios. To acquire a 1024-bit RIS during the decryption phase, we first use the attacked image as the extraction network's input. We then process the network's output using a voting technique. The encrypted position was obtained by applying the inverse RIS-generating procedure to the RIS during the decryption position identification stage. The attacked image was decrypted using the acquired secret keys and the encrypted position to generate the decrypted image. Since the extracted technique combines steganography with image encryption, we contrast its robustness, security, and imperceptibility with those of other schemes [34, 40]. We evaluate the performance of various schemes using metrics such as the peak signal-to-noise ratio (PSNR), mean-square error (MSE), and SSIM. The MSE metric is employed to quantify the discrepancy in quality between the original image and the processed image. The formula for calculating the MSE is as follows:

$$\text{MSE} = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - I'(i, j)]^2, \quad (11)$$

where I and I' are the original image and the noise image with $m \times n$ pixels, respectively.

The PSNR metric is employed to quantify the ratio between the quality of the original image and the quality of the processed image. The formula for calculating it is as follows:

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{\text{MAX}_I^2}{\text{MSE}} \right), \quad (12)$$

where MAX_I^2 is the maximum possible pixel value of the image.

Figure 5 shows the output results of the proposed scheme at each stage. It can be seen that the sensitive region in the image has been encrypted, and the sensitive region has been destroyed, resulting in visual unreadability, so the scheme has good security. The proposed approach offers strong imperceptibility, since we are unable to distinguish visually between the encrypted and embedded images. The proposed approach is reversible, because, following JPEG compression, the right RIS is extracted to determine the encryption position, the attacked image is decrypted with the correct key, and the sensitive region is precisely restored. Additionally, the proposed system can encrypt multiple sensitive regions, sensitive regions of varying sizes, and sensitive regions that are either regular or irregular. Overall, this trial demonstrated the efficacy of our image security system.

4.2 Security Analysis of the Key Space

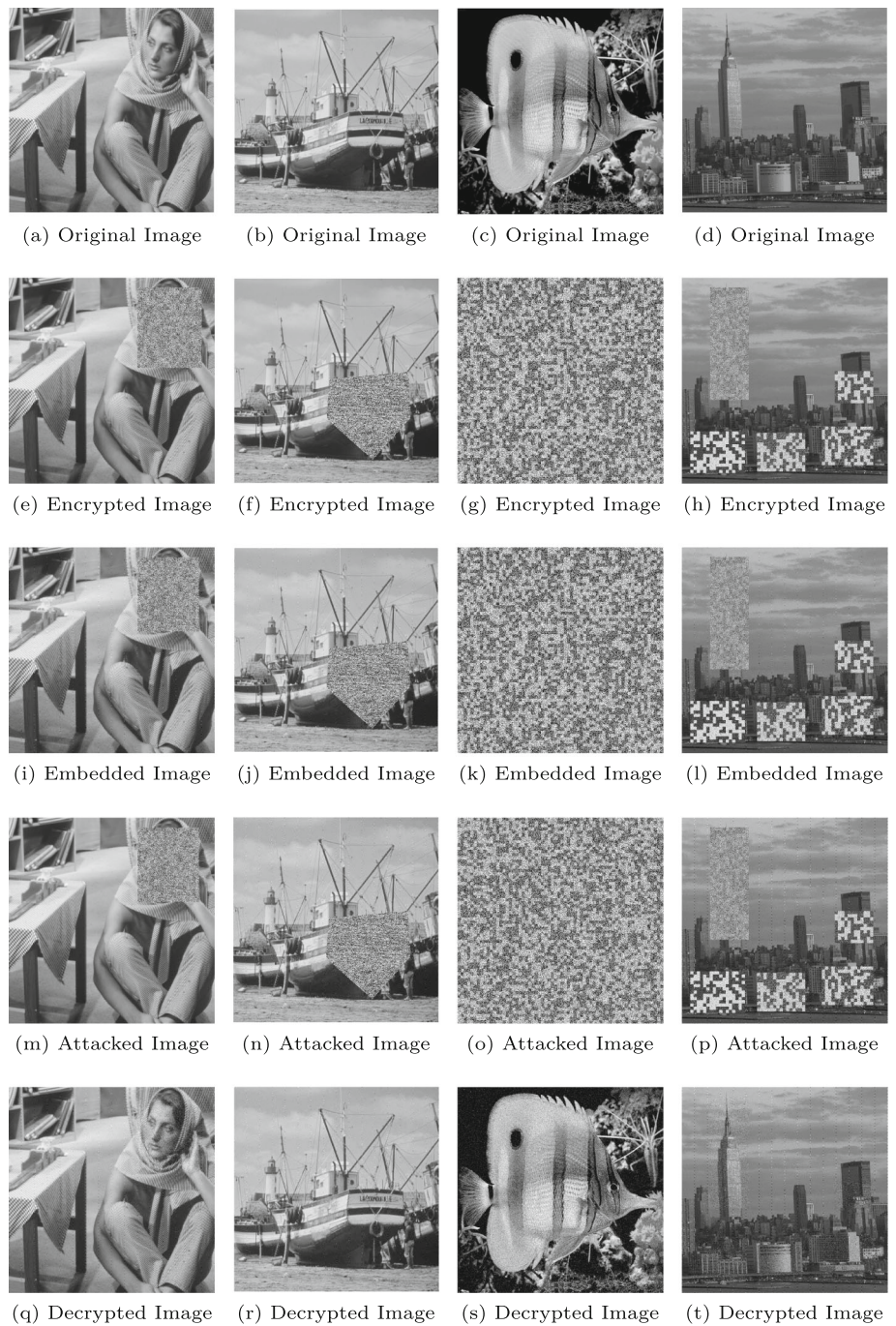
4.2.1 Key Space

The magnitude of the key space is a critical determinant in assessing the resilience of an encryption system against brute-force attacks. During such attacks, malevolent people systematically test every potential key within the encryption algorithm's key space to decipher the encrypted image. Increasing the size of the key space greatly increases the difficulty for malevolent users in deciphering the encryption key. This attack style is widely recognized as a brute-force attack. The size of the key space should be no less than 2^{128} to attain an elevated level of security to ensure a robust cryptosystem [41].

In the proposed technique, we establish a maximum precision of 14, resulting in a key space N_P of 10^{14} for block preprocessing. The key space N_S for block scrambling (step 2) is defined by the total number of possible permutations of n blocks, which can be stated as follows:

$$N_S = P_n^n = n! \quad (13)$$

Fig. 5 The output image of each step of the proposed scheme



Similarly, the key spaces for the other steps are given below

$$N_I = 4^n, N_R = 4^n, N_{I\&R} = 8^n, N_N = 2^n, \tag{14}$$

where N_I is the key space of block inversion, N_R is the key space of block rotation, $N_{I\&R}$ is the key space that combines block inversion and block rotation, and N_N is the key space of negative-positive transformation. Therefore, the key space

N_{total} of the proposed scheme is as follows:

$$\begin{aligned} N_{total} &= N_P \times N_S \times N_{I\&R} \times N_N \\ &= 10^{14} \times n! \times 8^n \times 2^n \\ &\approx 2^{4n+42} \times n! \end{aligned} \tag{15}$$

In our experiment, we utilized the original image dimensions of 512×512 , resulting in one of the trials containing $2^{18} 8 \times 8$ pixel blocks. Thus, the key space in our experiment exceeds 2^{200} , and we compare the experimental key space

Table 4 The key spaces for different solutions

Algorithm	Key space
Proposed	$> 2^{200}$
Content-selected image encryption algorithm [25]	2^{186}
Color image encryption technique [24]	2^{116}
Color image encryption algorithm [42]	2^{185}
Novel control method [43]	2^{152}
Image compression encryption hybrid algorithm [44]	2^{160}

with the actual scheme in Table 4. The proposed approach possesses a sufficiently extensive secret key space to effectively withstand exhaustive attacks.

4.2.2 Key Sensitivity

An encryption scheme should demonstrate strong key sensitivity to resist differential assaults. Consequently, even small alterations to the encryption key during the decryption procedure should produce discernible distortion in the decrypted image, thereby making the original form indistinguishable. This attribute guarantees that even slight deviations in the key produce substantial consequences, hence intensifying the challenge for attackers endeavoring to decipher the key and augment the overall security of the system. We made small adjustments to the parameters of the original key (U_1 , U_2 , ρ , ξ , and δ). As demonstrated in Fig. 6, for modifications 10^{-10} and 1 of ξ , ρ , U_1 , U_2 and δ , respectively, the decrypted RoS section is still full of noise. This is because the proposed scheme's secret key is based on a chaotic system, and due to the chaotic system's extremely sensitive features, even a slight disturbance will cause the subsequent secret key to entirely alter and fail to decrypt successfully. The encryption position is also one of the secret keys, as even with the secret key, it can be properly decrypted only if the proper encryption position is found. After moving the column of the encrypted position slightly to the right, we use the appropriate secret key to decrypt the data. The fact that the sensitive region is still not correctly recoverable, as depicted in the image, suggests that the proposed technique is key-sensitive.

4.3 Imperceptibility Analysis of Embedded Images

4.3.1 Analysis of Correlations

Image encryption techniques aim to improve image security by successfully breaking the correlation between adjacent pixels in the original image. In an original image, neighboring pixels usually show strong diagonal, vertical, and horizontal connections. Therefore, an efficient encryption algorithm should provide an encrypted image with the least correlation

possible between adjacent pixels along each of these three axes. For example, Fig. 7 shows a significant interpixel correlation since neighboring pixels in the original image tend to cluster along diagonal lines. Upon encryption, the neighboring pixels in the encrypted image are randomly distributed throughout space, effectively breaking the strong correlation between image pixels. This is because the proposed method increases the complexity of the encrypted image by performing a negation-positive transform operation on the secret region. An attacker finds it challenging to gather details about the original image from the encrypted image in order to reconstruct the original image because of this correlation-breaking effect. As a result, the proposed scheme can strengthen image confidentiality by using encryption.

4.3.2 Analysis of Invisibility

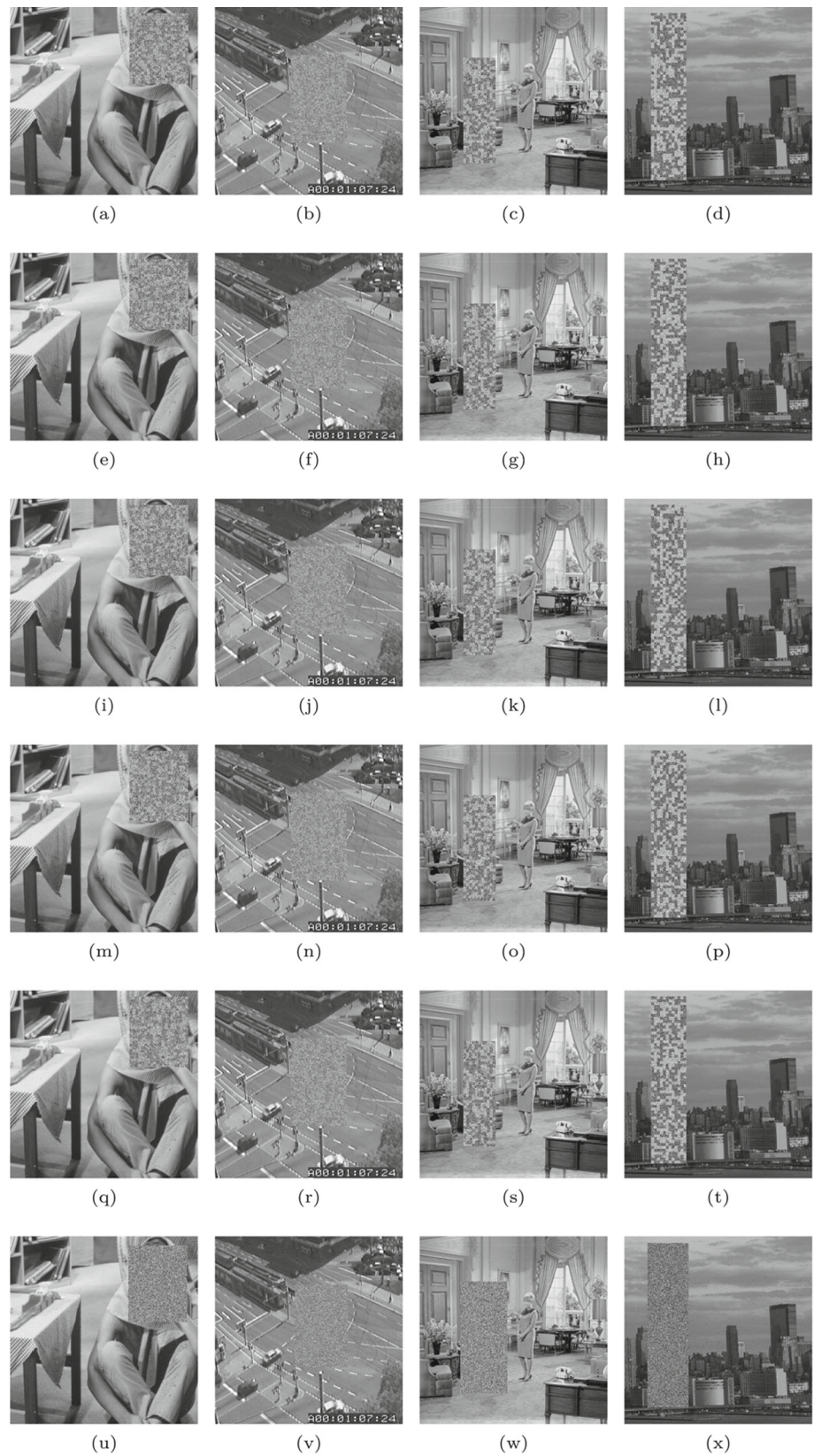
Imperceptibility, robustness, and embedding rate are crucial to a successful data concealment technique. Imperceptibility in real-world applications that require security, such as medical imaging, is particularly important. The embedding procedure produces an imperceptible embedded image, ideally having the same visual quality as the cover image, to prevent easy detection of the embedded information. Since the RIS is embedded inside the encrypted image as embedded information, the proposed approach requires that the embedded image maintain a substantial degree of visual similarity with the encrypted image. Specifically, to avoid information detection, the histogram of the embedded image should closely mirror the histogram of the encrypted image. After a detailed examination of the three test images and the corresponding histograms displayed in Fig. 8, it becomes evident that the encrypted and embedded histograms are similar. The end-to-end training framework effectively avoids detection of any hidden information and offers good data concealment due to the advancements in deep learning-based data-hiding technologies. This suggests that the scheme has good imperceptibility and successfully maintains the visual similarity between these two images.

Simultaneously, an analysis of the imperceptibility of the proposed technique is necessary to prevent third parties from identifying the RIS concealed within the embedded image. Our tests involve the detection of many indicators, including NC, PSNR, MSE, SSIM, and information entropy (IE). NC is utilized to identify the presence of patterns in two images. As the NC increases, so does the resemblance between the two images. The expression for NC is as follows:

$$NC = \frac{\sum A(i, j)B(i, j)}{\sqrt{\sum A(i, j)^2} \sqrt{\sum B(i, j)^2}}, \quad (16)$$

where A and B represent two different images. Moreover, IE is a quantitative measure used to assess the level of uncer-

Fig. 6 Key sensitivity analysis of disturbances of the secret key and position parameters: **a–d** decrypted images of $\xi - 10^{-10}$; **e–h** decrypted images of $\rho - 10^{-10}$; **i–l** decrypted images of $\delta - 1$; **m–p** Decrypted images of $U_1 - 10^{-10}$; **q–t** decrypted images of $U_2 - 10^{-10}$; **u–x** decrypted images of one pixel moved by 1 column



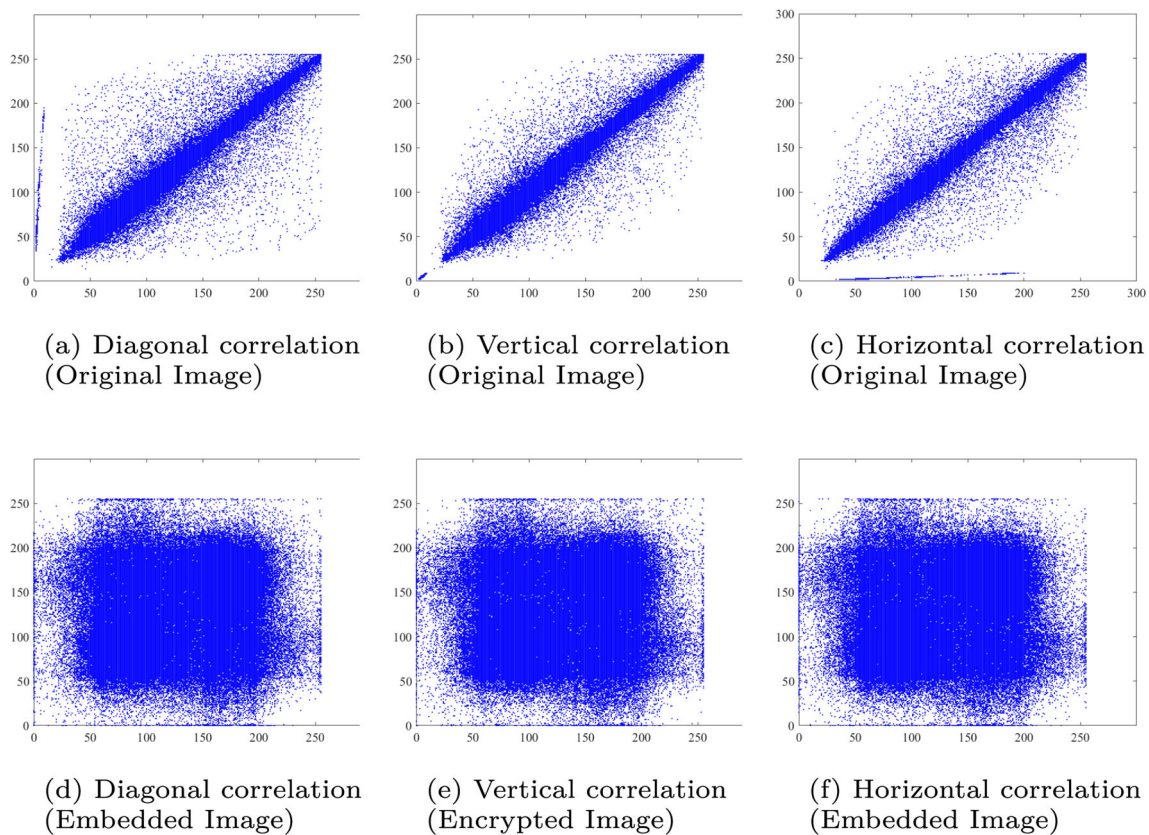


Fig. 7 The horizontal, vertical, and diagonal correlation between original and embedded image

tainty associated with an event, typically expressed in bits. The definition of the term is as follows:

$$IE = \sum P(A(i, j)) \log_2 P(A(i, j)), \quad (17)$$

where $P(A(i, j))$ is the probability of $A(i, j)$ occurring. According to Table 5, the proposed scheme has a high SSIM, indicating high similarity between the encrypted and embedded images. Additionally, the embedded image has good visual quality, with a PSNR greater than 33. The proposed method also produces similar IE for the embedded and encrypted images. Overall, the proposed scheme has decent imperceptibility.

4.4 Security Analysis of Embedded Images

4.4.1 Robustness Against Plaintext Attacks

The defense against attacks is where robustness is primarily seen. Among the most frequent types of network attacks, ciphertext-only attacks use the features of other encryption systems or the ciphertext itself to gather information about the original data without decrypting the ciphertext. We use an improved jigsaw solver to simulate ciphertext-only attacks to assess the robustness of the proposed system

[45, 46]. After being exposed to ciphertext-only assaults, the encrypted images of various schemes provide the information displayed in Fig. 9. The proposed scheme assembly outcomes provide less information than the typical scheme [34], but they also do not allow for the effective extraction of any information from the scheme. It is difficult for the puzzle solver to splice pixel blocks based on color, since the proposed scheme processes the image using grayscale operations, which significantly diminishes the continuity between pixels by decreasing the number of color channels in the image. In parallel, we apply a pixel scrambling procedure to the image, which further obliterates pixel relationships and dramatically enhances the difficulty of the puzzle solver. Generally, the proposed scheme can fend off attacks targeting ciphertext.

4.4.2 Robustness to JPEG Compression

The most popular lossy image compression technique in public networks, known as JPEG, reduces some of the original image's information during compression assaults. Robustness against JPEG compression is strongly needed in the proposed technique, since it is difficult to extract the whole RIS and properly identify the encrypted position to decrypt

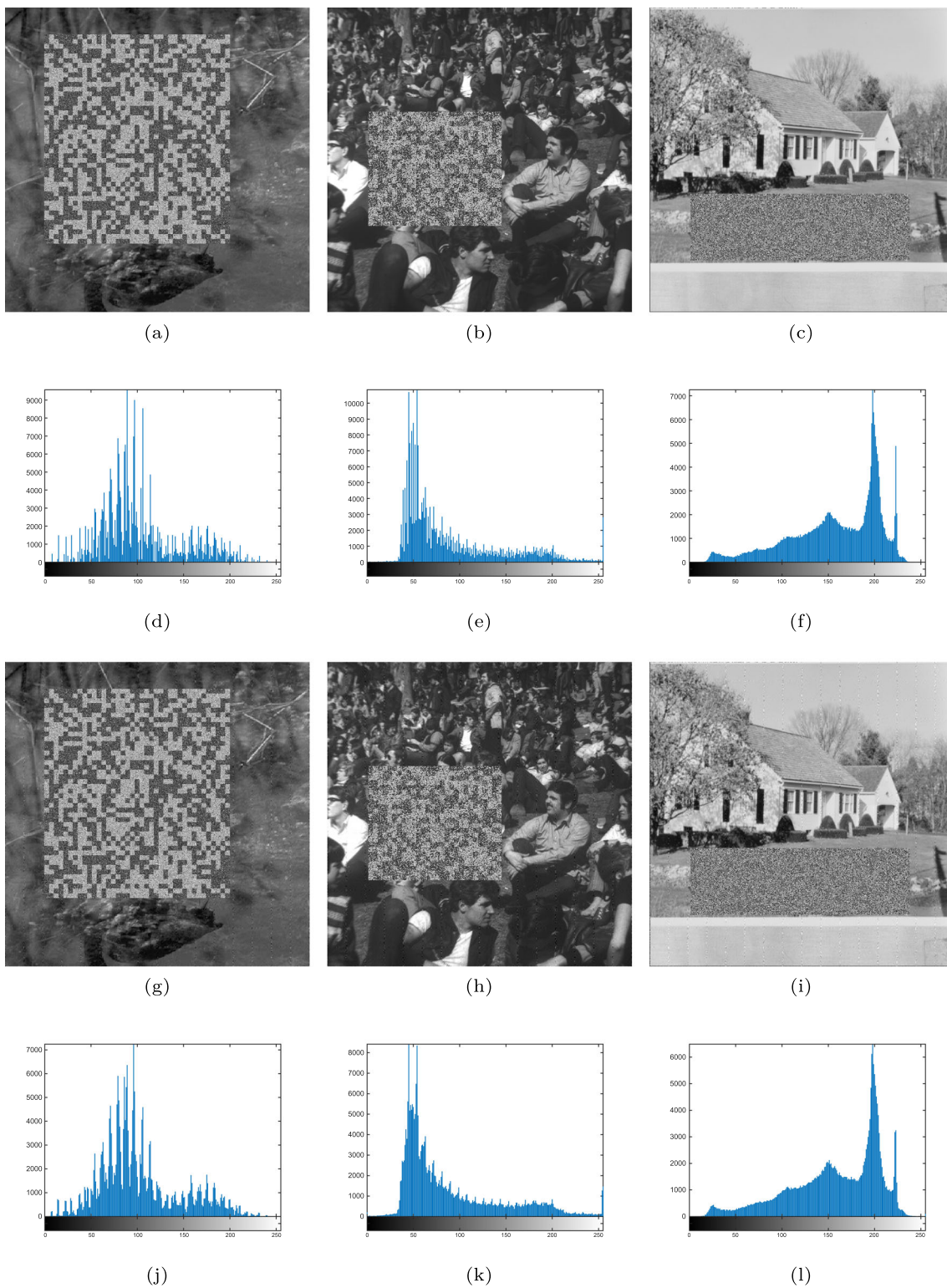


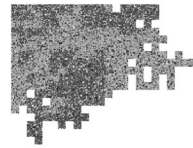
Fig. 8 Quality tests between different encrypted images and their corresponding embedded images

Table 5 The visual quality after JPEG compression of different parts of the image with different quality factors

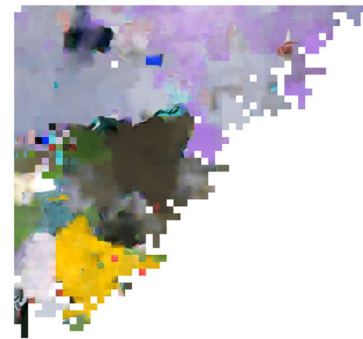
Test image	Information entropy		SSIM	NC	PSNR	MSE
	Encrypted image	Embedded image				
Animal	6.9212	7.1229	0.9494	0.9987	33.18	31.26
Human	6.9274	7.1083	0.9400	0.9985	33.01	32.54
House	7.2804	7.3225	0.9396	0.9995	33.40	29.71



(a) Original Image



(b) Assembled Image (proposed scheme)



(c) Assembled Image (EtC scheme)

Fig. 9 Images assembled from encrypted images via different schemes

the image. The proposed scheme's bit error rate (BER) decreases more quickly than does the comparative scheme's [33] when α increases, as Fig. 11 illustrates the robustness of various methods of JPEG compression. The proposed scheme's BER may be less than 0.05 of the comparative scheme when the α value is greater than 0.6. This is because the proposed system includes a 2×2 and 3×3 circular convolutions, which, in contrast to the comparison scheme, further encourages the diffusion and exchange of RIS data among image blocks. α is in charge of striking a balance between robustness and imperceptibility; when α is set to 0.6, the proposed scheme's BER is zero compared to the compared designs. The proposed scheme meets the robustness requirements without requiring α to be 1, which is different from the comparison schemes. This leads to an enhanced visual quality of the embedded image.

We set the strength factor α to 0.6 to obtain both good imperceptibility and robustness. The entire image and RoS following JPEG decryption subject to various quality variables are displayed in Fig. 10. When the quality factor decreases, a portion of the image's information is lost. The proposed method still offers respectable visual image quality and can be successfully decrypted. Table 6 displays the visual quality degradation curves of both the entire image and RoS following JPEG compression, allowing for a comparison of the two's image quality. Because RoS is encrypted, its visual quality is generally lower than that of the entire image, and it is more susceptible to JPEG compression than the entire

image. Overall, the proposed method may guarantee robustness and obtain a respectable level of image quality.

Table 7 displays the BER values of the data recovered by various schemes following JPEG compression to demonstrate the advantages of the proposed scheme in aiming for JPEG compression. Only the proposed schemes can achieve a BER value of 0 under three JPEG compressions, whereas many schemes can achieve very small BER values or even 0 after receiving JPEG compression with quality factors of 40, 30, and 20, respectively. This suggests that even with very low JPEG compression, the proposed scheme can still transmit safely in the channel. Nevertheless, the BER value of the proposed scheme is still much lower than that of other examined schemes, despite the fact that it cannot reach 0 when Q is 10. This indicates that the proposed scheme has exceptional robustness in JPEG compression (Fig. 11).

5 Conclusion and Future Work

This paper proposes an optimized image security technique that can both secure private information about an image and allow for image recompression and accurate identification, ensuring security and robustness throughout the transmission process. The secret region is encrypted using a block scrambling method based on a chaotic model to guarantee privacy and security. The chaotic model is responsible for preprocessing the irregular secret regions and producing more sensitive keys. Because of block scrambling, the

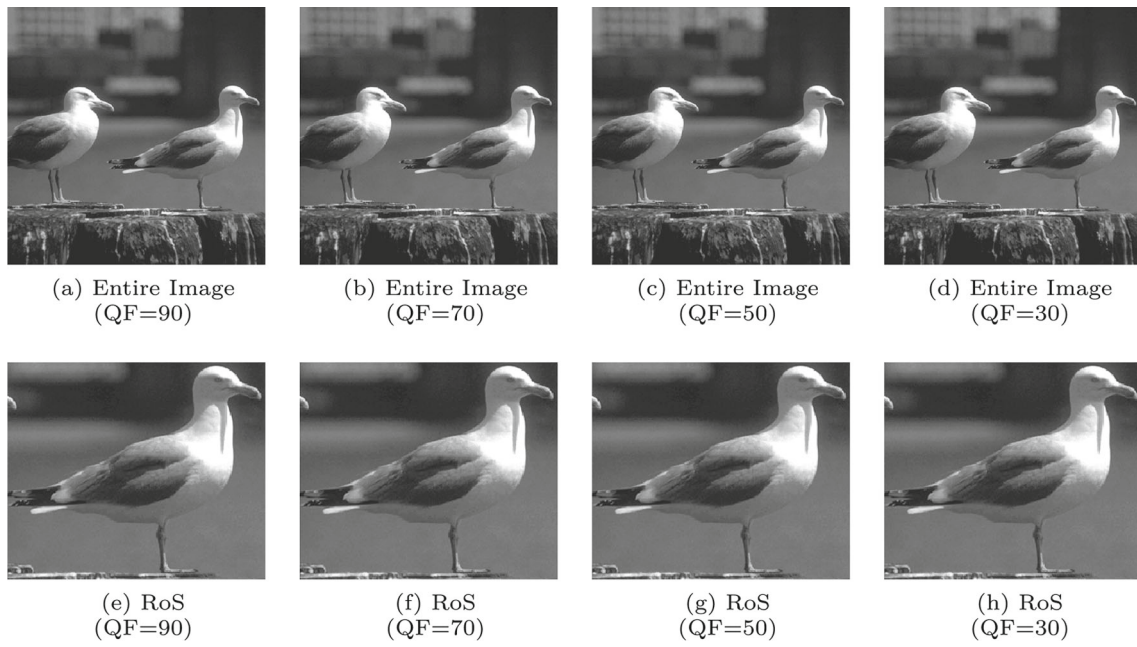


Fig. 10 Decrypted images and RoSs after JPEG compression with different compression factors

Table 6 The visual quality after JPEG compression of different parts of the image with different quality factors

Quality factor	Entire image			Region of secret		
	PSNR	MSE	SSIM	PSNR	MSE	SSIM
90	33.1	31.9	0.974	31.7	44.4	0.973
80	32.3	38.0	0.958	28.6	90.6	0.942
70	31.5	46.1	0.933	26.9	131.6	0.942
60	31.4	47.3	0.907	26.4	150.2	0.920
50	29.8	67.3	0.880	23.9	266.2	0.815
40	29.3	75.9	0.855	23.2	312.3	0.797
30	29.5	72.1	0.812	24.8	215.8	0.748

Table 7 BER values from different schemes and the proposed scheme under JPEG compression attacks

Algorithm	QF = 40	QF = 30	QF = 20	QF = 10
Wang’s scheme [47]	0.0259	0.0314	0.0333	0.0396
Zhang’s scheme [48]	0.0215	0.0293	0.0371	0.0507
Amini’s scheme [49]	0.0600	0.0609	0.0697	0.0732
Barlaskar’s scheme [50]	0.9776	0.9749	0.9666	0.9951
Shi’s scheme [51]	0	0.0261	0.0361	0.0517
Fan’s scheme [52]	0	0.0035	0.0407	0.3370
Ernawan’s scheme [53]	0.1553	0.2432	0.5010	0.5010
Jamali’s scheme [54]	0.0200	0.1500	0.3800	0.6200
Proposed’s scheme	0	0	0	0.0137

The bold meaning is to emphasize that BER can be 0 after the specified JPEG compression of the scheme. This means that the secret information embedded in the image can be correctly extracted

technique is successful against JPEG compression. Deep learning-based steganography not only makes the scheme less complex by allowing one to identify the secret region without the need for extra tools or information but also

increases the scheme’s security by requiring the extraction of the correct embedded data. The proposed scheme is tested in terms of JPEG compression, ciphertext-only assault, secret key space and sensitivity, and image quality. The experimen-

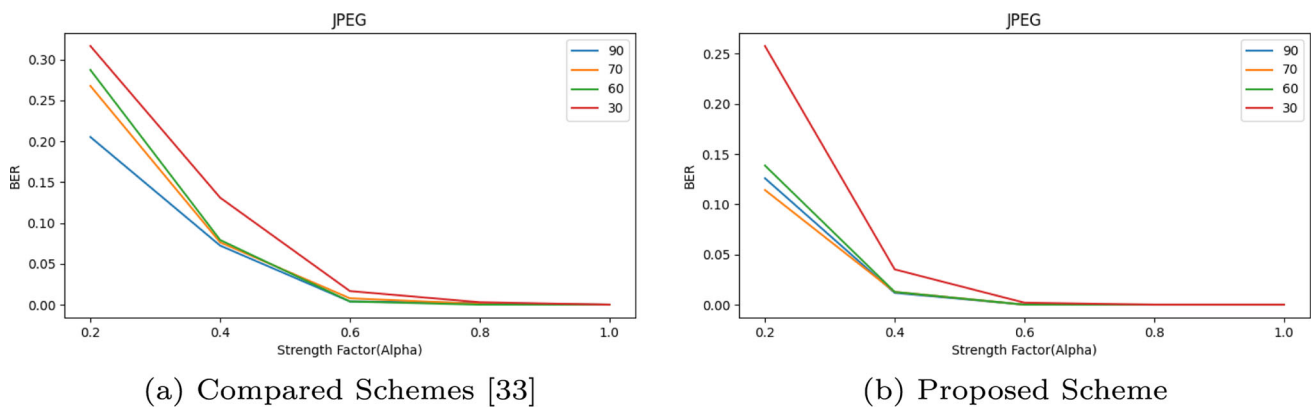


Fig. 11 Robustness of different schemes to JPEG compression

tal findings demonstrate that the proposed method is highly sensitive to secret key fine-tuning, has a sufficiently large secret key space to resist brute-force attacks, and has good imperceptibility. Effective information cannot be constructed following ciphertext-only attacks in terms of attack resistance, and the proposed approach can still retain a BER of 0 and correctly decrypt the image even in the face of JPEG compression with a low quality factor. Furthermore, we verify the superiority of the proposed scheme by conducting a comparative analysis with the conventional schemes. A comparative analysis reveals that the proposed scheme outperforms the current schemes with respect to the ciphertext-only attack, secret key space and sensitivity, and JPEG compression.

The proposed scheme has various limitations, including restricted embedding capacity, limited grayscale images, and low visual quality of images, even though it offers greater security and robustness in data transmission. Thus, to achieve improved performance on public networks, future research should concentrate on creating high-capacity embedding frameworks for RGB images. Future research should concentrate on employing image-denoising networks to enhance the visual quality of images and minimize the effects of JPEG compression on images.

Author Contributions ZC: conceptualization, methodology, and writing original draft. YL: validation and review. GK: validation and review. JW: validation and review. WZ: review and writing. SL: validation, writing, editing, and supervision.

Funding This research was funded by the Macau Science and Technology Development Funds [Grant No. 0061/2020/A2].

Data Availability Statement We evaluate our method on the public Pascal VOC2012, CIFAR10, Fdez-Vidal, Cub200-2011, and PatternNet datasets. The Pascal VOC2012, CIFAR10, Fdez-Vidal, and Cub200-2011 datasets are available at <http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html>, <http://www.cs.toronto.edu/protect/unhbox/voidb@xpenalty/@M/kriz/cifar.html>, <https://ccia.>

ugr.es/cvg/CG/base.htm, https://www.vision.caltech.edu/datasets/cub_200_2011/, respectively.

Declarations

Conflict of Interest No conflict of interest exists in the submission of this manuscript, and the manuscript has been approved by all the authors for publication. I would like to declare on behalf of my coauthors that the work described was original research that has not been published previously and is not under consideration for publication elsewhere, in whole or in part. All the authors listed have approved the enclosed manuscript.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Huang, C.-T., et al.: Survey on securing data storage in the cloud. *APSIPA Trans. Signal Inf. Process.* **3**, 7 (2014)
- Hua, H., et al.: Edge computing with artificial intelligence: a machine learning perspective. *ACM Comput. Surv.* **55**(9), 1–35 (2023)
- Almalawi, A., et al.: A hybrid cryptographic mechanism for secure data transmission in edge AI networks. *Int. J. Comput. Intell. Syst.* **17**(1), 24 (2024)
- Chuman, T., et al.: Security evaluation for block scrambling-based image encryption including jpeg distortion against jigsaw puzzle solver attacks. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **101**(12), 2405–2408 (2018)
- Kamal, A.A.A.M., et al.: Searchable encryption of image based on secret sharing scheme. In: 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), pp. 1495–1503 (2017)

6. Cheng, H., Zhang, X., Yu, J.: AC-coefficient histogram-based retrieval for encrypted jpeg images. *Multimed. Tools Appl.* **75**(21), 13791–13803 (2016)
7. Xu, Y., Gong, J., et al.: A privacy-preserving content-based image retrieval method in cloud environment. *J. Vis. Commun. Image Represent.* **43**, 164–172 (2017)
8. Zhou, J., Liu, X., Au, O.C., Tang, Y.Y.: Designing an efficient image encryption-then-compression system via prediction error clustering and random permutation. *IEEE Trans. Inf. Forensics Secur.* **9**(1), 39–50 (2013)
9. Chuman, T., Sirichotedumrong, W., Kiya, H.: Encryption-then-compression systems using grayscale-based image encryption for jpeg images. *IEEE Trans. Inf. Forensics Secur.* **14**(6), 1515–1525 (2019)
10. Sirichotedumrong, W., Kiya, H.: Grayscale-based block scrambling image encryption using YCBCR color space for encryption-then-compression systems. *APSIPA Trans. Signal Inf. Process.* **8**, 7 (2019)
11. Kurihara, K., Imaizumi, S., et al.: An encryption-then-compression system for lossless image compression standards. *IEICE Trans. Inf. Syst.* **100**(1), 52–56 (2017)
12. Hu, G., Peng, J., Kou, W.: A novel algorithm for generating pseudo-random number. *Int. J. Comput. Intell. Syst.* **12**(2), 643–648 (2019)
13. Indira, N., Rukmanidevi, S., Kalpana, A.: Light weight proactive padding based crypto security system in distributed cloud environment. *Int. J. Comput. Intell. Syst.* **13**(1), 36–43 (2020)
14. Abdullah, S., Ayub, S., et al.: Analyses of S-boxes based on interval valued intuitionistic fuzzy sets and image encryption. *Int. J. Comput. Intell. Syst.* **10**(1), 851–865 (2017)
15. Johnson, M., Ishwar, P., et al.: On compressing encrypted data. *IEEE Trans. Signal Process.* **52**(10), 2992–3006 (2004)
16. Schonberg, D., Draper, S.C., Ramchandran, K.: On blind compression of encrypted data approaching the source entropy rate. In: 2005 13th European Signal Processing Conference, pp. 1–4 (2005)
17. Pinto, M., et al.: Protection of jpeg compressed e-comics by selective encryption. In: 2013 IEEE International Conference on Image Processing, pp. 4588–4592 (2013)
18. Chen, Y.-H., Lu, E.J.-L., Wang, C.-F.: Privacy image protection using fine-grained mosaic technique. In: 2013 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, pp. 1–4 (2013)
19. Wen, W., et al.: A novel selective image encryption method based on saliency detection. In: 2016 Visual Communications and Image Processing (VCIP), pp. 1–4 (2016)
20. Khan, J.S., Ahmad, J.: Chaos based efficient selective image encryption. *Multidimens. Syst. Signal Process.* **30**(2), 943–961 (2019)
21. Ju, Q., Xia, R., et al.: Privacy-preserving classification on deep learning with exponential mechanism. *Int. J. Comput. Intell. Syst.* **17**(1), 39 (2024)
22. Goel, S., et al.: Image steganography–least significant bit with multiple progressions. In: Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA), vol. 328, pp. 105–112 (2015)
23. Gupta, M., Gupta, K.K., et al.: Session key based fast, secure and lightweight image encryption algorithm. *Multimed. Tools Appl.* **80**(7), 10391–10416 (2021)
24. Hosny, K.M., et al.: A color image encryption technique using block scrambling and chaos. *Multimed. Tools Appl.* **81**(1), 505–525 (2022)
25. Wang, J., Liu, L., et al.: A novel content-selected image encryption algorithm based on the LS chaotic model. *J. King Saud Univ.-Comput. Inf. Sci.* **34**(10), 8245–8259 (2022)
26. Ping, P., Zhang, X., et al.: A novel medical image encryption based on cellular automata with ROI position embedded. *Multimed. Tools Appl.* **81**(5), 7323–7343 (2022)
27. Ding, Q., Ding, Q., Perrizo, W.: Parm—an efficient algorithm to mine association rules from spatial data. *IEEE Trans. Syst. Man Cybern. B (Cybern.)* **38**(6), 1513–1524 (2008)
28. Luo, T., et al.: Stereo matching based stereo image watermarking for tamper detection and recovery. *Int. J. Comput. Intell. Syst.* **7**(5), 874–881 (2014)
29. Chen, J., et al.: Cryptographic secrecy analysis of matrix embedding. *Int. J. Comput. Intell. Syst.* **6**(4), 639–647 (2013)
30. Anbarjafari, G., Ozcinar, C.: Imperceptible non-blind watermarking and robustness against tone mapping operation attacks for high dynamic range images. *Multimed. Tools Appl.* **77**(18), 24521–24535 (2018)
31. Yang, Y., Lei, M., et al.: A robust blind audio watermarking scheme based on singular value decomposition and neural networks. *Int. J. Comput. Intell. Syst.* **7**(5), 865–873 (2014)
32. Sinhal, R., et al.: Real-time watermark reconstruction for the identification of source information based on deep neural network. *J. Real-Time Image Proc.* **17**(6), 2077–2095 (2020)
33. Ahmadi, M., et al.: Redmark: framework for residual diffusion watermarking based on deep networks. *Expert Syst. Appl.* **146**, 113157 (2020)
34. Iida, K., Kiya, H.: An image identification scheme of encrypted jpeg images for privacy-preserving photo sharing services. In: 2019 IEEE International Conference on Image Processing (ICIP), pp. 4564–4568 (2019)
35. Huang, X., et al.: Visually meaningful image encryption algorithm based on digital signature. *Digit. Commun. Netw.* **9**(1), 159–165 (2023)
36. Abadi, M., et al.: Tensorflow: a system for large-scale machine learning. In: 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), pp. 265–283 (2016)
37. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. <http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html> (2012)
38. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images. *Handb. Syst. Autoimmun. Dis.* **1**(4), 32–33 (2009)
39. Fdez-Vidal, X. R. (b): Compound gain: Visual distinctness metric for coder performance evaluation. <http://decsai.ugr.es/cvg/CG/base.htm> (2014)
40. Chen, Y.-H., Lu, E.J.-L., Wang, C.-F.: Light-weight selective image encryption for privacy preservation. *J. Electron. Sci. Technol.* **18**(3), 234–240 (2020)
41. Li, R., Liu, Q., Liu, L.: Novel image encryption algorithm based on improved logistic map. *IET Image Proc.* **13**(1), 125–134 (2019)
42. Liu, Q., Liu, L.: Color image encryption algorithm based on DNA coding and double chaos system. *IEEE Access* **8**, 83596–83610 (2020)
43. Chen, C., Sun, K., et al.: A novel control method to counteract the dynamical degradation of a digital chaotic sequence. *Eur. Phys. J. Plus* **134**, 1–16 (2019)
44. Zhang, Y., Xu, B., Zhou, N.: A novel image compression-encryption hybrid algorithm based on the analysis sparse representation. *Opt. Commun.* **392**, 223–233 (2017)
45. Chuman, T., Kurihara, K., Kiya, H.: On the security of block scrambling-based ETC systems against jigsaw puzzle solver attacks. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2157–2161 (2017)
46. Chuman, T., Kurihara, K., Kiya, H.: Security evaluation for block scrambling-based ETC systems against extended jigsaw puzzle solver attacks. In: 2017 IEEE International Conference on Multimedia and Expo (ICME), pp. 229–234 (2017)
47. Wang, X., Shen, X., Tian, J.-L., Niu, P., Yang, H.: Statistical image watermark decoder based on local frequency-domain exponent-

- Fourier moments modeling. *Multimed. Tools Appl.* **80**(18), 27717–27755 (2021)
48. Wang, X.-Y., et al.: Locally optimum image watermark decoder by modeling NSCT domain difference coefficients with vector based Cauchy distribution. *J. Vis. Commun. Image Represent.* **62**, 309–329 (2019)
49. Amini, M., et al.: A robust multibit multiplicative watermark decoder using a vector-based hidden Markov model in wavelet domain. *IEEE Trans. Circuits Syst. Video Technol.* **28**(2), 402–413 (2016)
50. Barlaskar, S.A., et al.: Genetic algorithm based optimized watermarking technique using hybrid DCNN-SVR and statistical approach for watermark extraction. *Multimed. Tools Appl.* **81**(5), 7461–7500 (2022)
51. Shi, H., et al.: A novel zero-watermarking algorithm based on multi-feature and DNA encryption for medical images. *Multimed. Tools Appl.* **82**, 36507–36552 (2023)
52. Fan, M.: Blind dual image watermarking for copyright protection, tamper proofing and self-recovery. *Multimed. Tools Appl.* **82**, 45503–45518 (2023)
53. Ernawan, F., Kabir, M.N.: A robust image watermarking technique with an optimal DCT-psychovisual threshold. *IEEE Access* **6**, 20464–20480 (2018)
54. Jamali, M., et al.: Adaptive blind image watermarking using fuzzy inference system based on human visual perception. *arXiv preprint [arXiv:1709.06536](https://arxiv.org/abs/1709.06536)* (2017)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.