Nonlinear Dyn (2021) 104:2759–2805
https://doi.org/10.1007/s11071-021-06409-z

ORIGINAL PAPER

An efficient dual-layer cross-coupled chaotic map security-based multi-image encryption system

K. Abhimanyu Kumar Patro · Bibhudendra Acharya

Received: 8 October 2020 / Accepted: 24 March 2021 / Published online: 7 April 2021

Abstract This paper proposes an effective image encryption method to encrypt several images in one encryption process. The proposed encryption scheme differs from the current multiple image encryption schemes because of their two-layer cross-coupled chaotic map-based permutation-diffusion operation. Block-shuffling, left–right (L–R) flipping and then bit-XOR diffusion operations are carried out in the first layer using one set of cross-coupled chaotic map. Block-shuffling, up–down (U–D) flipping and then bit-XOR diffusion operations are performed with another set of cross-coupled chaotic maps in the second layer. The two different layers of permutation-diffusion make the proposed algorithm more efficient than the existing multi-image encryption algorithms. Moreover, the combination of block-based shuffling and flip operation decreases the algorithm's computational complexity, which means enhancing the time efficiency of the algorithm. In cross-coupling operation, the use of a single fixed-type one-dimensional chaotic map-piece-wise linear chaotic map (PWLCM) makes the algorithm efficient both in hardware and in software. PWLCM's initial values and system parameters (keys) are generated by means of the hash values of original images that resist the algorithm against the attacks of chosen-plaintext and known-plaintext. Results of simulation and comparative security analysis reveal that the suggested scheme is more effective in encryption and resists better against all widely used attacks.

Keywords Security · Multi-image encryption · Cross-coupling · Data hiding · Chaos · Hash algorithm

1 Introduction

Through technological advances, the transmission of information over public networks has grown rapidly. The information is conveyed either in the form of videos, audios, text, or images, and nowadays many of the relevant information is represented by images. Images are widely used in military secrets, satellite reports, government secret documents, etc. Security of images therefore becomes a major factor. Encryption is one of the common approaches to image security. There are several conventional methods (DES [1], 3DES, AES [2], RSA, etc.) for encrypting images. Nevertheless, due to large information and strong association of neighboring pixels in the images, these approaches are not as efficient for encrypting images

K. A. K. Patro · B. Acharya (✉)
Department of Electronics and Communication Engineering, National Institute of Technology Raipur, Raipur, Chhattisgarh 492010, India
e-mail: bacharya.etc@nitrr.ac.in

K. A. K. Patro
e-mail: kakpatro.etc@nitrr.ac.in

[3, 4]. Hence, the need for a strong encryption method is of highest concern.

Encryption methods based on chaos offer significant protection for image encryption. The distinctive characteristics of chaotic systems like ergodicity, mixing properties, sensitivity to chaotic variables, and highly complex behavior make it an ideal choice for a powerful and efficient encryption method [5–7]. Many single-image-encryption (SIE) methods based on DNA-level [5, 8–10], block-level [11, 12], pixel-level [3, 6, 13–15], transform operation-based pixel-level [16], rotation operation-based pixel-level [17], hash-key operation-based pixel-level [18], s-box operation-based pixel-level [19], bit-level [4, 7], bit-block-level [20], permutation and diffusion operation employ chaotic maps to build secure and strong cipher images. The types of chaotic systems employed in SIE methods are "one dimensional" (1D) and "high dimensional" (HD) [13]. Chaotic 1D maps tend to be powerful, are simpler in structure and thus use resources more effectively, but they have the disadvantage of a limited key space. Chaotic HD maps have a wide range of key spaces, but they are highly complex and thus more resource utilization [14, 20]. So, to provide the algorithm with high efficiency and strong key space, in image encryption algorithms [15, 17] authors have used multi-type 1D chaotic maps multiple times. But using different kinds of chaotic 1D maps reduces the algorithm's hardware and software efficiency. To resolve this problem, the suggested method uses the PWLCM system multiple times for encryption operation.

During the age of big data, the communication network transmits multiple images. SIE methods can be repetitively used in the encryption to protect those images. However, they reduce the efficiency of encryption [21]. In order to boost the encryption efficiency, many researchers use multiple image encryption (MIE) algorithms to encrypt multiple images. From the past few years, in different domains the MIE algorithms are developed such as transform [22–24], chaotic [25–27], mix of transform and chaotic [28], and mix of chaotic and DNA domain [29, 30].

In the above MIE algorithms, the transform domain-based MIE algorithms [22–24, 28] decrease the algorithm's encryption efficiency. This is because there is a need for information translation among transform and spatial domain within these algorithms. The MIE algorithms based on the chaotic domain boost the algorithm's encryption efficiency. In Ref. [25], Tang et al. developed a bit-plane process-based MIE algorithm. This algorithm uses the chaotic map to execute the encryption process; however, the complex bit-plane process decreases the algorithm's encryption efficiency [26]. In Ref. [27], Zhang et al. developed a method to increase the encryption efficiency in a chaotic map-based MIE algorithm. No doubt this algorithm increases the efficiency of encryption, but the security is still lower than Ref. [25]'s algorithm. This is because the position of blocks and its contents are processed in the algorithm of Ref. [25], while the ordering of the blocks is processed only in the algorithm of Ref. [27]. Hence, the security of the algorithm in Ref. [27] is reduced. Furthermore, the algorithm's secret keys in Ref. [27] do not link to the actual images. This may be due to the possibility of the chosen-plaintext and known-plaintext being targeted. To resolve all of the above issues, another MIE technique was developed in Ref. [26] by the same author. This technique maintains the security as well as increases the algorithm's encryption efficiency. To enhance the security in MIE algorithms, many researchers use DNA along with chaotic maps to encrypt multiple images. References [29] and [30] developed MIE algorithms using DNA and chaotic maps. However, the additional DNA processes like encoding–decoding enhance the algorithm's computational complexity.

In the above chaotic map-based MIE algorithms [25–27, 29, 30], one common issue is that the encrypted image relies only a single chaotic system chaotic orbit, so in those algorithms it may be possible to extract information [31]. This fact has led to the development of many cryptanalysis algorithms [32–36] to cryptanalyze analog security communication methods. When using the chaotic map cross-coupling, such cryptanalysis will be harder. This is

because different chaotic orbits of cross-coupled chaotic maps determine the resultant cipher output [31, 37, 38]. Recently, in 2020, Patro et al. [39] developed a chaotic map cross-coupling-based MIE method. In this method, the chaotic map cross-coupling-based permutation-diffusion operation ensures security to multiple images. Here the two different PWLCM systems are used for cross-coupling operation. There is no question that the algorithm in Ref. [39] provides security, but the security is either not much improved or compromised in some cases. The key space of the method in Ref. [39] is not larger than that of the method of Ref. [25]. As we observed the algorithm's histogram variance analysis and Chi-square test analysis in Ref. [39], we found that the grayscale pixel values of Group-2 images are less uniformly distributed as compared with Group-1 images. That means the algorithm in Ref. [39] is not equally performed to all the groups of images. As we found in the entropy analysis of the algorithm in Ref. [39], we notice that, relative to the current MIE algorithms, the average entropy of both image groups is not increased.

By addressing all of the above problems, this paper develops an effective multi-image encryption technique based on two layers of cross-coupled chaotic map. Block permutation, left–right (L–R) flip operation and bit-XOR diffusion are performed in the first layer of operation, and block permutation, up–down (U–D) flip operation and bit-XOR diffusion process are executed again in the second layer of operation. Two different sets of cross-coupled chaotic maps are used in the first and second layers of permutation-diffusion operation to enhance the security. The advantages of the proposed scheme are as follows.

- Uses cross-coupling of chaotic maps to permute and diffuse the pixels. Cryptanalysis of the proposed algorithm is harder as compared to the algorithms using non-cross-coupling of chaotic maps to permute and diffuse the pixels.
- In cross-coupling, the use of the only PWLCM system improves the algorithm's hardware and software performance as compared to the algorithms using multiple types of other 1D and HD chaotic maps in permutation and diffusion operations. The PWLCM system improves the algorithm's encryption speed also. It is because the PWLCM system's single iteration process requires multiple additions and a single division only.
- The performance of both block- and pixel-level permutation operations improves the security of the proposed algorithm as compared to the algorithms performing either block-level or pixel-level permutation operations.
- In the proposed algorithm, dual-layer security exists. This implies that the proposed algorithm performs two-time two-way block and pixel-level permutation operations and two-times XOR-based image diffusion operations. The other multiple image encryption algorithms, on the other hand, only carry out one layer of security.
- In the proposed algorithm, flip operation (L–R and U–D)-based pixel permutation is performed to permute the pixels. Flip processes in current CPU architectures are completed in just a few clock cycles. Flipping processes do not require constant time; it is one of the fastest processes in current CPU architectures. On the other hand, the other multiple image encryption algorithms perform pixel-permutation operations, offering an $O(M \times N)$ computational complexity, where $M \times N$ is the image size.

Based on the aforementioned considerations, this paper's key contribution is as given below.

- Two layers of cross-coupling operation are performed to make the algorithm more effective.
- To make the encryption scheme both hardware and software efficient, in cross-coupling process, a single chaotic map-PWLCM is used.
- Block-based permutation and flip operation are implemented to reduce the algorithm's computational complexity.
- Use of hash-based keys protects the algorithm against the attacks of chosen- and known-plaintext.

The remainder of the paper is structured according to the following. The PWLCM system is introduced in Sect. 2. Section 3 describes the approach suggested here. The algorithm's security analysis and the outcomes of the simulation are discussed in Sect. 4. At last, this paper concludes in Sect. 5.

2 PWLCM system

In image encryption, the two important characteristics of chaotic maps such as "simplicity" and "ergodicity" must be considered for selection of any chaotic map. The map that provides both "simplicity" and broader "ergodicity" is PWLCM. The logistic map, on the other side, is the popular chaotic map for "simplicity" but has no wider "ergodicity". So the best choice is the PWLCM system for the generation of good pseudo-random chaotic sequences. It is generated by [26, 39],

$$k_{n+1} = \begin{cases} \frac{k_n}{m} & \text{if} \quad 0 \le k_n < m \\ \frac{k_n - m}{0.5 - m} & \text{if} \quad m \le k_n < 0.5 \\ (1 - k_n) & \text{if} \quad 0.5 \le k_n < 1 \end{cases} \tag{1}$$

where the initial value $k \in (0, 1)$ and the system parameter $m \in (0, 0.5)$. Reference [39] shows the bifurcation diagrams in two figures. The bifurcation of the logistic map is seen in one figure (Fig. 1), and the bifurcation of PWLCM system is seen in another figure (Fig. 2). By analyzing these two figures, it is noticed that the ergodicity of logistic map exists in the range $(0, 1)$ when the system parameter m reaches 4, while the ergodicity of PWLCM system exists in a wider range. Hence, in many image encryption algorithms, PWLCM system is preferred. Besides ergodicity and simplicity, the other significant characteristics of the PWLCM system are easy software and hardware realization, good dynamic behavior, uniform invariant distribution and effective implementation [39, 40].

3 Proposed methodology

3.1 Method of key generation operation

1. Take L images with the same $M_g \times N_g$ dimension.
2. Merge horizontally with all the L images. The image that is generated being referred by $L1$.
3. Apply the hash algorithm, SHA-256 to image $L1$ to produce the hash value of 256-bit. It is depicted by

   $$hvb = (hvb_1, hvb_2, hvb_3, \ldots, hvb_{256})$$

4. Convert the hash value of 256 bit into the hash value of 64-hexadecimal. Figure 1 shows the process of conversion of binary to hexadecimal. In this process, the entire 256 bits are split into 64 groups, each of which contains 4 bits. Each of the 4-bit groups will then be translated to its corresponding hexadecimal number.
   The 64-hexadecimal hash value is represented as

   $$hvh = (hvh_1, hvh_2, hvh_3, \ldots, hvh_{64})$$

5. Develop the algorithm's keys by,

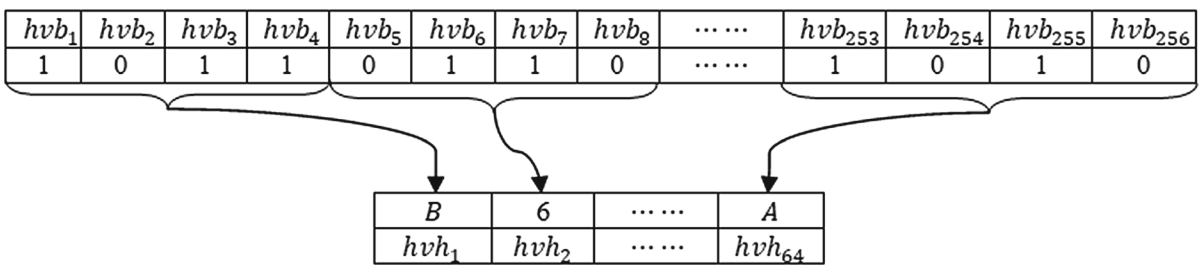


Fig. 1 Binary-to-hexadecimal conversion process

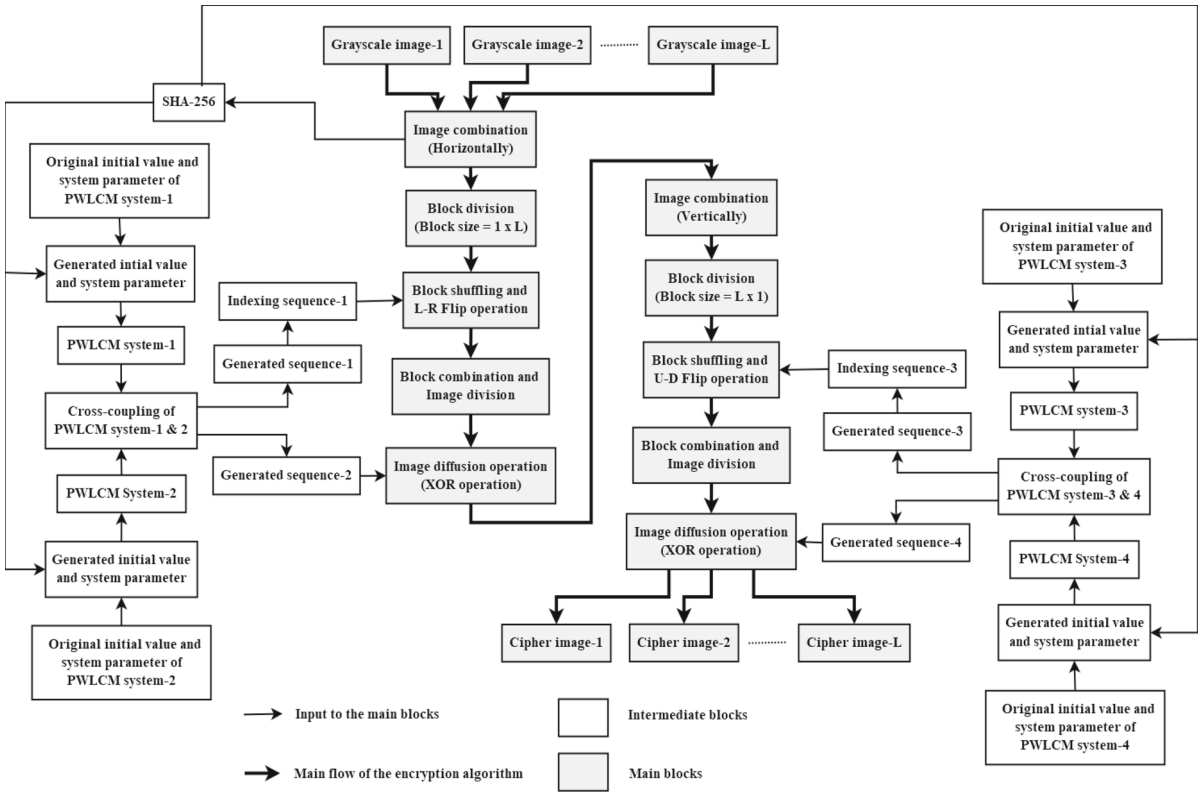


Fig. 2 Block schematic representation of the suggested encryption scheme

$$\begin{cases} m(1) = km - \left(\left(\frac{hvh_1 + hvh_2 + \cdots + hvh_8}{10^{15}} \right) - \left\lceil \frac{hvh_1 + hvh_2 + \cdots + hvh_8}{10^{15}} \right\rceil \right) \times 0.01 \\ mk = mm - \left(\left(\frac{hvh_9 + hvh_{10} + \cdots + hvh_{16}}{10^{15}} \right) - \left\lceil \frac{hvh_9 + hvh_{10} + \cdots + hvh_{16}}{10^{15}} \right\rceil \right) \times 0.01 \end{cases} \tag{2}$$

$$\begin{cases} n(1) = kn - \left(\left(\frac{hvh_{17} + hvh_{18} + \cdots + hvh_{24}}{10^{15}} \right) - \left\lceil \frac{hvh_{17} + hvh_{18} + \cdots + hvh_{24}}{10^{15}} \right\rceil \right) \times 0.01 \\ nk = nn - \left(\left(\frac{hvh_{25} + hvh_{26} + \cdots + hvh_{32}}{10^{15}} \right) - \left\lceil \frac{hvh_{25} + hvh_{26} + \cdots + hvh_{32}}{10^{15}} \right\rceil \right) \times 0.01 \end{cases} \tag{3}$$

$$\begin{cases} p(1) = kp - \left(\left(\frac{hvh_{33} + hvh_{34} + \cdots + hvh_{40}}{10^{15}} \right) - \left\lceil \frac{hvh_{33} + hvh_{34} + \cdots + hvh_{40}}{10^{15}} \right\rceil \right) \times 0.01 \\ pk = pp - \left(\left(\frac{hvh_{41} + hvh_{42} + \cdots + hvh_{48}}{10^{15}} \right) - \left\lceil \frac{hvh_{41} + hvh_{42} + \cdots + hvh_{48}}{10^{15}} \right\rceil \right) \times 0.01 \end{cases} \tag{4}$$

$$\begin{cases} q(1) = kq - \left(\left(\frac{hvh_{49} + hvh_{50} + \cdots + hvh_{56}}{10^{15}} \right) - \left\lceil \frac{hvh_{49} + hvh_{50} + \cdots + hvh_{56}}{10^{15}} \right\rceil \right) \times 0.01 \\ qk = qq - \left(\left(\frac{hvh_{57} + hvh_{58} + \cdots + hvh_{64}}{10^{15}} \right) - \left\lceil \frac{hvh_{57} + hvh_{58} + \cdots + hvh_{64}}{10^{15}} \right\rceil \right) \times 0.01 \end{cases} \tag{5}$$

where

$(m(1), n(1), p(1), q(1))$ and (km, kn, kp, kq) are the PWLCM system-1- to 4-based generated and given initial values, respectively.

(mk, nk, pk, qk) and (mm, nn, pp, qq) are the PWLCM system-1- to 4-based generated and given system parameters, respectively.

The symbol $\lceil . \rceil$ performs the "*ceil*" operation.

3.2 Method of encryption operation

Figure 2 presents a block schematic representation of the suggested encryption scheme. The steps for the encryption process are as follows.

1. Take L images $(I_1, I_2, I_3, \ldots, I_L)$ with the same $M_g \times N_g$ dimension.
2. Merge horizontally with all the L images by,

   $$L1 = horzcat(I_1, I_2, I_3, \ldots, I_L)$$

   where $L1$ is the horizontal concatenated image. The function "*horzcat*" performs the horizontal concatenation operation. The size of $L1$ is $M_g \times LN_g$.
3. Generate blocks of size $1 \times L$ by dividing the image $L1$. The number of generated blocks is $\frac{M_g \times LN_g}{1 \times L} = M_g \times N_g$. The block division process is presented in Algorithm 1.
   In the above algorithm, $B1$ is the array of blocks of size $1 \times L$.
4. Calculate the number of blocks in $B1$ by,

   $$l1 = length(B1)$$

   where $l1$ is the length of the array $B1$. The function *"length"* finds the length of an array.
5. Generate the PWLCM system-1 and system-2-based keys by Eqs. (2) and (3), respectively. $(m(1), mk)$ and $(n(1), nk)$ are the PWLCM system-1- and system-2-based generated keys, respectively.
6. Generate second initial values of PWLCM system-1 and system-2 using key values $(m(1), mk)$ and $(n(1), nk)$, respectively, to iterate Eq. (1). The second initial values are denoted as $m(2)$ of PWLCM system-1 and $n(2)$ of PWLCM system-2.
7. Perform cross-coupling operation $(M_g \times N_g) - 1$ times between the key values $(m(2), nk)$ and $(n(2), mk)$. Algorithm 2 shows the process of cross-coupling operation.

Algorithm 1

Input : Horizontally concatenated image $L1$ and the size of $L1$.
Output: Array of blocks $B1$.

```
s1 ← 1
for i1 ← 1 to M_g by an increment of 1 do
    for j1 ← 1 to LN_g by an increment of L do
        B1{s1} ← L1(i1, j1: j1 + L − 1)
        s1 ← s1 + 1
    end
end
```

Algorithm 2

Input : Key values $(m(2), nk)$ and $(n(2), mk)$.
Output: Iterated sequence m & n of PWLCM system-1 & 2, respectively.

```
for i2 ← 2 to (M_g × N_g) − 1 by an increment of 1 do
    if n(i2) ≥ 0 and n(i2) < mk do
        m(i2 + 1) ← n(i2)/mk
    else if n(i2) ≥ mk and n(i2) < 0.5 do
        m(i2 + 1) ← (n(i2) − mk)/(0.5 − mk)
    else if n(i2) > 0.5 and n(i2) < 1 do
        if 1 − n(i2) ≥ 0 and 1 − n(i2) < mk do
            m(i2 + 1) ← (1 − n(i2))/mk
        else if 1 − n(i2) ≥ mk and 1 − n(i2) < 0.5 do
            m(i2 + 1) ← (1 − n(i2) − mk)/(0.5 − mk)
        end
    end
    if m(i2 + 1) ≥ 0 and m(i2 + 1) < nk do
        n(i2 + 1) ← m(i2 + 1)/nk
    else if m(i2 + 1) ≥ nk and m(i2 + 1) < 0.5 do
        n(i2 + 1) ← (m(i2 + 1) − nk)/(0.5 − nk)
    else if m(i2 + 1) > 0.5 and m(i2 + 1) < 1 do
        if 1 − m(i2 + 1) ≥ 0 and 1 − m(i2 + 1) < nk do
            n(i2 + 1) ← (1 − m(i2 + 1))/nk
        else if 1 − m(i2 + 1) ≥ nk and 1 − m(i2 + 1) < 0.5 do
            n(i2 + 1) ← (1 − m(i2 + 1) − nk)/(0.5 − nk)
        end
    end
end
```

The iterated sequences are represented as,

$$m = \left(m(1), m(2), m(3), \ldots, m\left(M_g \times N_g\right)\right)$$

$$n = \left(n(1), n(2), n(3), \ldots, n\left(M_g \times N_g\right)\right)$$

8. Sort the iterated sequence m by,

$$[permlrsort, permlrindex] = sort(m)$$

where *permlrindex* is the indexed sequence and *permlrsort* is the sorted sequence of m.

9. Perform block shuffling and left–right (L–R) flip operation between the blocks of the array $B1$. In the combined operation, first the blocks are L–R-flipped and then shifted to a new position in the array using the indexing sequence *permlrindex*. The newly shuffled array of blocks are denoted as $B11$.
10. Generate a big image $B111$ by combining the blocks of the array $B11$.
11. Divide the big image $B111$ into L parts. The newly formed images are denoted as $II_1, II_2, II_3, \ldots, II_L$.
12. Generate an image r by using the iterated sequence n (From Algorithm 2). Algorithm 3 presents the process to generate an image r.

Algorithm 3

Input : Iterated sequence n from **Algorithm 2**.
Output: Image r.

```
r ← round(n × 10^6)
r ← r mod 256
r ← reshape(r, [M_g, N_g])
```

In the above algorithm, r is the newly formed image. The *"reshape"* function re-shapes the array, the *"mod"* function executes the modulus process, and the *"round"* function rounding the number to the closest number.

13. Execute diffusion process (bit-XOR) among the

images $II_1, II_2, II_3, \ldots, II_L$ and the newly formed image r. The bit-XOR diffusion process is first carried out between II_1 and r, and then, the diffusion process is executed between II_2 and the first diffusion output. Likewise, all the $II_1, II_2, II_3, \ldots, II_L$ images are bit-XORed. The diffused images are denoted as $III_1, III_2, III_3, \ldots, III_L$.

14. Combine all the diffused images $III_1, III_2, III_3, \ldots, III_L$ vertically by,

$$LL1 = vertcat(III_1, III_2, III_3, \ldots, III_L)$$

where $LL1$ is the vertically concatenated image. The function "*vertcat*" performs the vertical concatenation operation. The size of $LL1$ is $LM_g \times N_g$.

15. Generate blocks of size $L \times 1$ by dividing the image $LL1$. The number of generated blocks are $\frac{LM_g \times N_g}{L \times 1} = M_g \times N_g$. The block division process is presented in Algorithm 4.

Algorithm 4

Input : Vertically concatenated image $LL1$ and the size of $LL1$.
Output: Array of blocks $B2$.

```
s2 ← 1
for i4 ← 1 to LM_g by an increment of L do
    for j4 ← 1 to N_g by an increment of 1 do
        B2{s2} ← LL1(i4: i4 + L − 1, i4)
        s2 ← s2 + 1
    end
end
```

In the above algorithm, $B2$ is the array of blocks of size $L \times 1$.

16. Calculate the number of blocks in $B2$ by,

$$l2 = length(B2)$$

where $l2$ is the $B2$ array length.

17. Generate the PWLCM system-3- and system-4-based keys by Eqs. (4) and (5), respectively. $(p(1), pk)$ and $(q(1), qk)$ are the PWLCM system-3- and system-4-based generated keys, respectively.

18. Generate second initial values of PWLCM system-3 and system-4 using key values $(p(1), pk)$ and $(q(1), qk)$, respectively, to iterate Eq. (1). The second initial values are denoted as $p(2)$ of PWLCM system-3 and $q(2)$ of PWLCM system-4.

19. Perform cross-coupling operation $(M_g \times N_g) - 1$ times between the key values $(p(2), qk)$ and $(q(2), pk)$. The cross-coupling process is carried out in the same way as Algorithm 2. The generated iterated sequences are represented as,

$$p = \big(p(1), p(2), p(3), \ldots, p\big(M_g \times N_g\big)\big)$$

$$q = \big(q(1), q(2), q(3), \ldots, q\big(M_g \times N_g\big)\big)$$

20. Sort the iterated sequence p by

$$[permudsort, permudindex] = sort(p)$$

where *permudindex* is the indexed sequence and *permudsort* is the sorted sequence of p.

21. Perform block shuffling and up–down (U–D) flip operation between the blocks of the array $B2$. In the combined operation, first the blocks are U–D-flipped and then shifted to a new position in the array using the indexing sequence *permudindex*. The newly shuffled array of blocks are denoted as $B222$.

22. Generate a big image $B222$ by combining the blocks of the array $B222$.

23. Divide the big image $B222$ vertically into L parts. The newly formed images are denoted as $III_1, III_2, III_3, \ldots, III_L$.

24. Generate an image rr by using the iterated sequence q. The image generation process is carried out in the same way as Algorithm 3.

25. Perform bit-XOR diffusion operation between the images $III_1, III_2, III_3, \ldots, III_L$ and the newly formed image rr. First the bit-XOR diffusion operation is performed between III_1 and rr, and then, the diffusion process is carried out among III_2 and the first diffusion output. Likewise, all the $III_1, III_2, III_3, \ldots, III_L$ images are bit-XORed. The diffused images are denoted as $IIII_1, IIII_2, IIII_3, \ldots, IIII_L$. These diffused images are the L cipher images denoted as $C_1, C_2, C_3, \ldots, C_L$.

3.3 Method of decryption operation

1. On the receiver side, the receiver collects all the cipher images $C_1, C_2, C_3, \ldots, C_L$ of size $M_g \times N_g$ produced on the transmitter side, given key values of the algorithm such as initial values (km, kn, kp, kq) of PWLCM system-1 to 4, system parameters (mm, nn, pp, qq) of PWLCM

system-1 to 4. Receiver also receives multi-image hash values.

2. Generate the PWLCM system-3- and system-4-based keys by Eqs. (4) and (5), respectively. $(p(1), pk)$ and $(q(1), qk)$ are the PWLCM system-3- and system-4-based generated keys, respectively.
3. Follow Sect. 3.2 Step-18 to produce PWLCM system-3-based second initial value using key values $(p(1), pk)$ and also produce PWLCM system-4-based second initial value using key values $(q(1), qk)$.
4. Follow Sect. 3.2 Step-19 to perform the cross-coupling operation $(M_g \times N_g) - 1$ times between the key values $(p(2), qk)$ and $(q(2), pk)$. The iterated sequences generated after cross-coupling are,

$$p = \big(p(1), p(2), p(3), \ldots, p\big(M_g \times N_g\big)\big)$$

$$q = \big(q(1), q(2), q(3), \ldots, q\big(M_g \times N_g\big)\big)$$

5. Follow Sect. 3.2 Step-24 to develop an image *crr* using the iterated sequence q.
6. Execute diffusion process (bit-XOR) among the cipher images $C_1, C_2, C_3, \ldots, C_L$ and the newly formed image *crr*. First the process of bit-XOR diffusion is performed between C_1 and *crr*. Second, the diffusion of bit-XOR occurs between C_1 and C_2. Third the diffusion between C_2 and C_3 is performed. Likewise, all the cipher images are diffused. The diffused cipher images are denoted as $CC_1, CC_2, CC_3, \ldots, CC_L$.
7. Follow Sect. 3.2 Step-14 to combine vertically all the diffused images $CC_1, CC_2, CC_3, \ldots, CC_L$. The image being concatenated vertically is denoted as CC. The size of CC is $LM_g \times N_g$.
8. Follow Sect. 3.2 Step-15 to segment the CC image into numbers of blocks of the same size $L \times 1$. The number of blocks generated is $\frac{LM_g \times N_g}{L \times 1} = M_g \times N_g$. The blocks are stored in an array which is called $CC1$.
9. Follow Sect. 3.2 Step-16 to calculate the array length of $CC1$. The array length is denoted as $l2$.
10. Follow Sect. 3.2 Step-20 to sort the iterated sequence p. The sorted and indexed sequences are denoted as *cpermudsort* and *cpermudindex*.
11. Perform block de-shuffling and up–down (U–D) flip operation between the blocks of the array $CC1$. In the combined operation, first the blocks are U–D flipped and then de-shuffled using the indexing sequence *cpermudindex*. The newly de-shuffled array of blocks are denoted as $CC2$.
12. Follow Sect. 3.2 Step-22 to create an image $CC3$ by combining the $CC2$ array blocks together.
13. Vertically divide the image $CC3$ into L parts. The newly formed images are denoted as $CC3_1, CC3_2, CC3_3, \ldots, CC3_L$.
14. Generate the PWLCM system-1 and system-2-based keys by Eqs. (2) and (3), respectively. $(m(1), mk)$ and $(n(1), nk)$ are the PWLCM system-1- and system-2-based generated keys, respectively.
15. Follow Sect. 3.2 Step-6 to generate the PWLCM system-1-based second initial value using key values $(m(1), mk)$ and also generate the PWLCM system-2-based second initial value using key values $(n(1), nk)$.
16. Follow Sect. 3.2 Step-7 to perform the cross-coupling operation $(M_g \times N_g) - 1$ times between the key values $(m(2), nk)$ and $(n(2), mk)$. The iterated sequences generated after cross-coupling are,

$$m = \big(m(1), m(2), m(3), \ldots, m\big(M_g \times N_g\big)\big)$$

$$n = \big(n(1), n(2), n(3), \ldots, n\big(M_g \times N_g\big)\big)$$

17. Follow Sect. 3.2 Step-12 to develop an image *cr* using the iterated sequence n.
18. Execute diffusion process (bit-XOR) among the images $CC3_1, CC3_2, CC3_3, \ldots, CC3_L$ and the newly formed image *cr*. First the process of bit-XOR diffusion is performed between $CC3_1$ and *cr*. Second, the diffusion of bit-XOR occurs between $CC3_1$ and $CC3_2$. Third the diffusion between $CC3_2$ and $CC3_3$ is performed. Likewise, all the images are diffused. The diffused images are denoted as $CCC3_1, CCC3_2, CCC3_3, \ldots, CCC3_L$.
19. Follow Sect. 3.2 Step-2 to combine horizontally all the diffused images $CCC3_1, CCC3_2, CCC3_3, \ldots, CCC3_L$. The image being concatenated horizontally is

Table 1 Original keys of the algorithm

Chaotic maps	System parameters	Initial values
First PWLCM	$mm = 0.356049176629853$	$km = 0.257841690334592$
Second PWLCM	$nn = 0.367403327981564$	$kn = 0.256107392758891$
Third PWLCM	$pp = 0.360919367233587$	$kp = 0.269879664352183$
Fourth PWLCM	$qq = 0.376915442905374$	$kq = 0.273128967510886$

Table 2 Hexadecimal hash values of two image groups

Image groups	Hash values
Group-1 ("Lena", "Baboon", "Pepper", and "Sailboat")	"07bb3bacbf7b373a4b9afffad8d7fe7b2c44269f232104a2387969d3dd9976e9"
Group-2 ("Elaine", "Baboon", "Boat", and "Couple")	"f75776c876597af69777b1e89b7daab83ae7fcdba896e9c503f4a861f15a3eea"

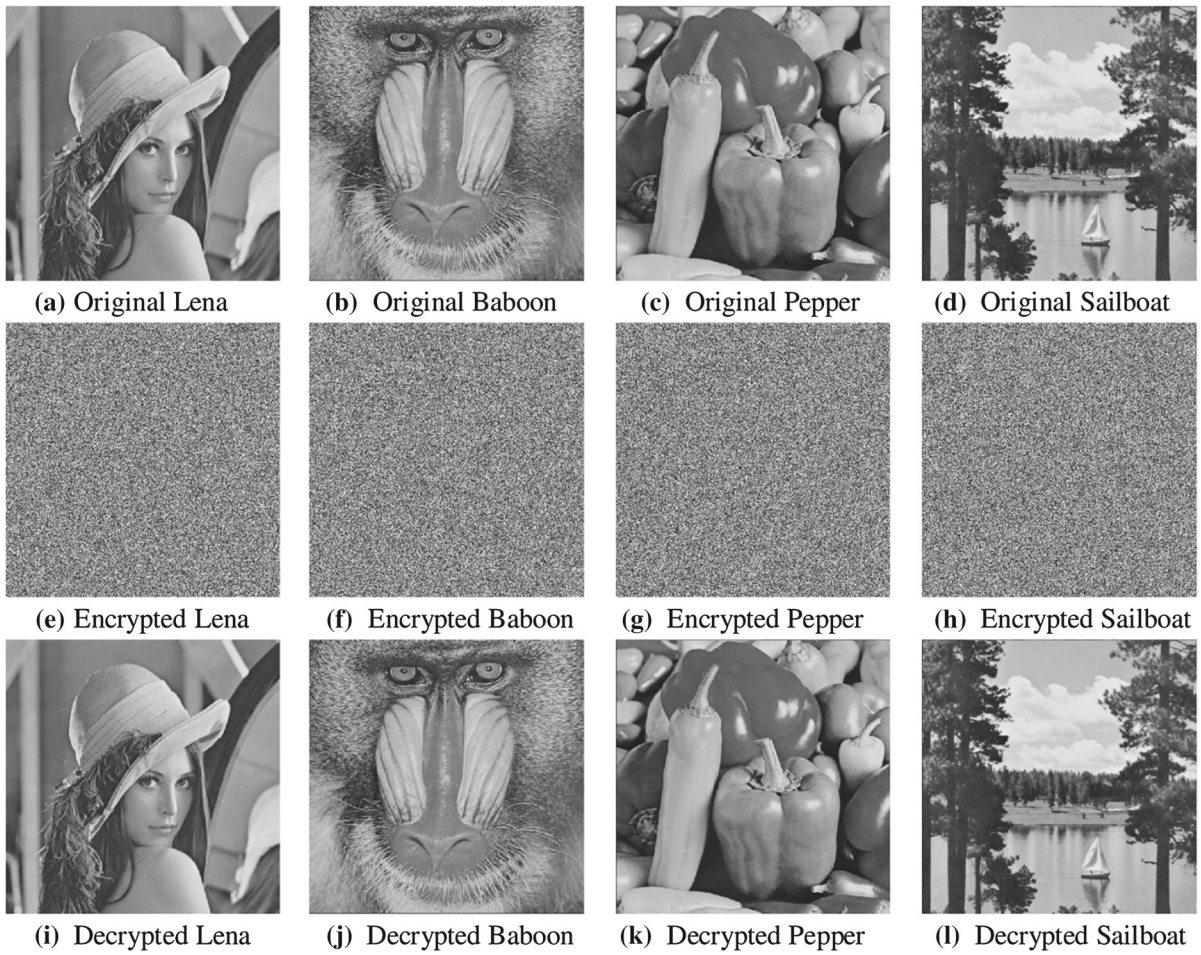


Fig. 3 Simulation outcomes of Group-1 images

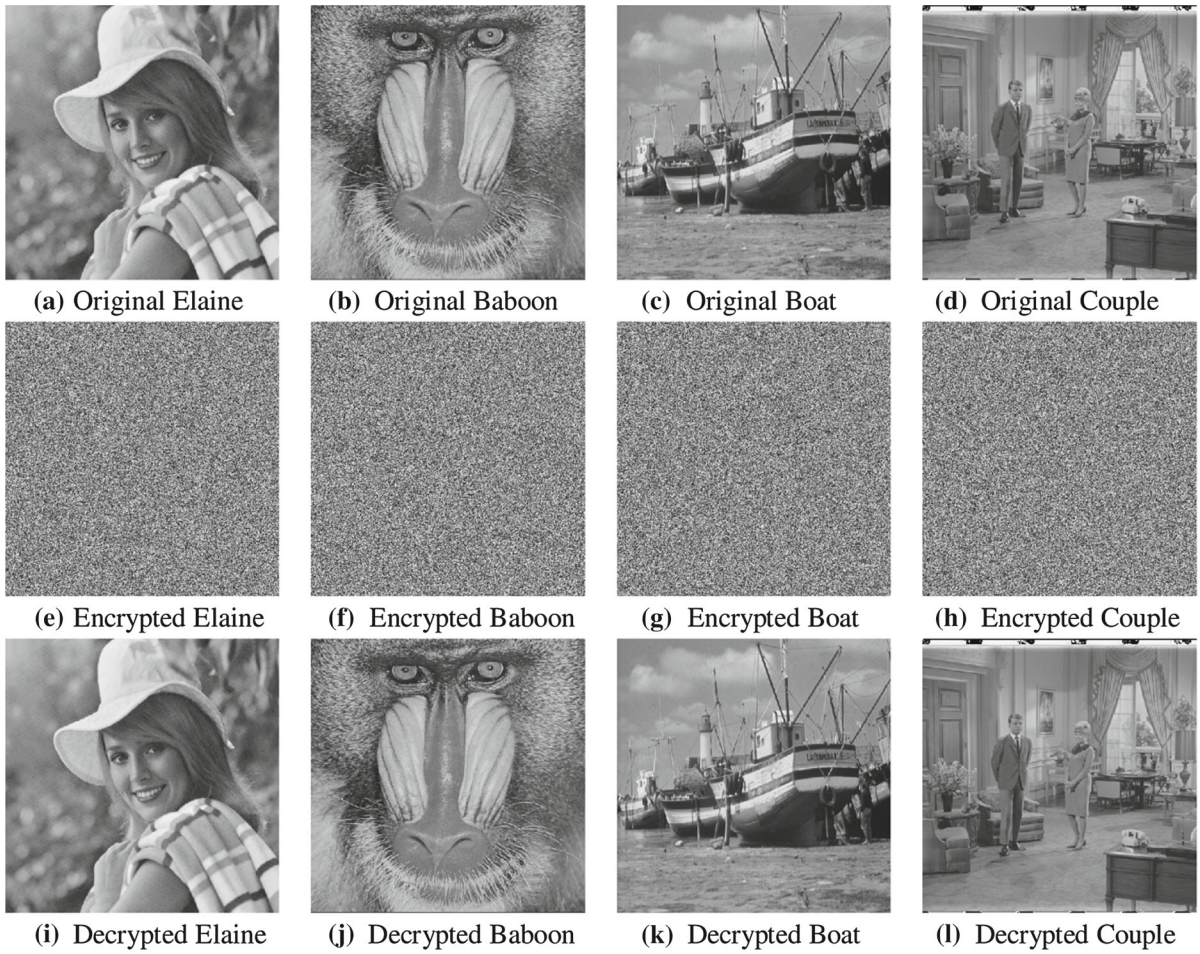


Fig. 4 Simulation outcomes of Group-2 images

denoted as $CCC3$. The size of $CCC3$ is $M_g \times LN_g$.

20. Follow Sect. 3.2 Step-3 to segment the $CCC3$ image into numbers of blocks of the same size $1 \times L$. The number of blocks generated is $\frac{M_g \times LN_g}{1 \times L} = M_g \times N_g$. The blocks are stored in an array, which is called $CCC2$.
21. Follow Sect. 3.2 Step-4 to calculate the array length $CCC2$. The array length is denoted as $l1$.

Table 3 Key space of the algorithm

Key parameters	Key space	Total key space
First PWLCM (km, mm)	$10^{15} \times 10^{15} = 10^{30}$	$10^{30} \times 10^{30} \times 10^{30} \times 10^{30} \times 2^{128} \approx 1.5491 \times 2^{526}$
Second PWLCM (kn, nn)	$10^{15} \times 10^{15} = 10^{30}$	
Third PWLCM (kp, pp)	$10^{15} \times 10^{15} = 10^{30}$	
Fourth PWLCM (kq, qq)	$10^{15} \times 10^{15} = 10^{30}$	
Hash values	2^{128}	

22. Follow Sect. 3.2 Step-8 to sort the iterated sequence m. The sorted and indexed sequences are denoted as *cpermlrsort* and *cpermlrindex*.
23. Perform block de-shuffling and left–right (L–R) flip operation between the blocks of the array $CCC2$. In the combined operation, first the blocks are L–R-flipped and then de-shuffled using the indexing sequence *cpermlrindex*. The newly de-shuffled array of blocks is denoted as $CCC1$.
24. Follow Sect. 3.2 Step-10 to create an image CCC by combining the $CCC1$ array blocks together.
25. Horizontally divide the image $CCC2$ into L parts. The newly formed images are denoted as $CCC_1, CCC_2, CCC_3, \ldots, CCC_L$. These images are the L original images.

4 Computer simulations and security analyses

In this paper, the simulation of the suggested scheme is carried out on two image groups such as Image Group-1 that includes "Lena", "Baboon", "Pepper", and "Sailboat", Image Group-2 that includes "Elaine", "Baboon", "Boat", and "Couple". Computer simulations are conducted on a system with a processor of 2.50 GHz, RAM of 4.00 GB using MATLAB. All images are 512×512 in size and are obtained from the database of USC-SIPI [41]. The keys given for the suggested scheme are listed in Table 1, and the hexadecimal equivalent hash values for two image groups are shown in Table 2. The simulation outcomes of Group-1 images are shown in Fig. 3, and Group-2 images are shown in Fig. 4. It is found in the simulation outputs that the encrypted images tend to be very noisy, indicating that attackers could not get any details from them about the original images. This illustrates that our method has a strong encryption effect. It is also found in the simulation outputs that we can get the successful decrypted images using the correct secret keys.

Table 4 Comparison of key space results

Encryption schemes	Total key space
Proposed	1.5491×2^{526}
Zhang and Wang	
Third scheme [29]	1.0195×2^{186}
Second scheme [26]	1.2446×2^{199}
First scheme [27]	1.0195×2^{186}
Tang et al. [25]	8.1148×2^{445}
Patro et al. [39]	1.2446×2^{327}

The suggested scheme also applies for the encryption of multiple color images. This can only be achieved through encrypting the Red (R), Green (G), and Blue (B) parts separately and then combining all of the encrypted R, G, and B parts to get multiple encrypted color images.

The proposed algorithm's security evaluation is as follows.

4.1 Key space analysis

Key space is the number of distinct keys used to execute the process of encryption [42]. An algorithm's key space must be greater than 2^{128} to withstand the attack of brute-force [8, 43]. The keys used in the suggested scheme are:

- the PWLCM system-1- to system-4 based initial values $km, kn, kp,$ andkq
- the PWLCM system-1- to system-4-based system parameters $mm, nn, pp,$ andqq
- Hash values of 256-bit

The algorithm's key space is presented in Table 3. In the table, it is shown that for each of the algorithm's individual keys, a key space of 10^{15}, is used. It is because the algorithm uses the 64-bit floating point standard and for that 10^{15} is suggested by IEEE [44]. The 2^{128} key space for the SHA-256 hash-value is also shown in the table. This is because the security in the hash function SHA-256 to withstand the best attack is 2^{128}. Hence, the total key space of the proposed algorithm shown in the table is 1.5491×2^{526}, which is greater than 2^{128} to effectively protect the attack of brute force.

The comparison of key space among the suggested method and the existing reference methods [25–27, 29, 39] is provided in Table 4. The results of the comparison indicate that the suggested method has larger key space than Refs. [25–27, 29, 39] methods. It indicates that the suggested method highly resists the

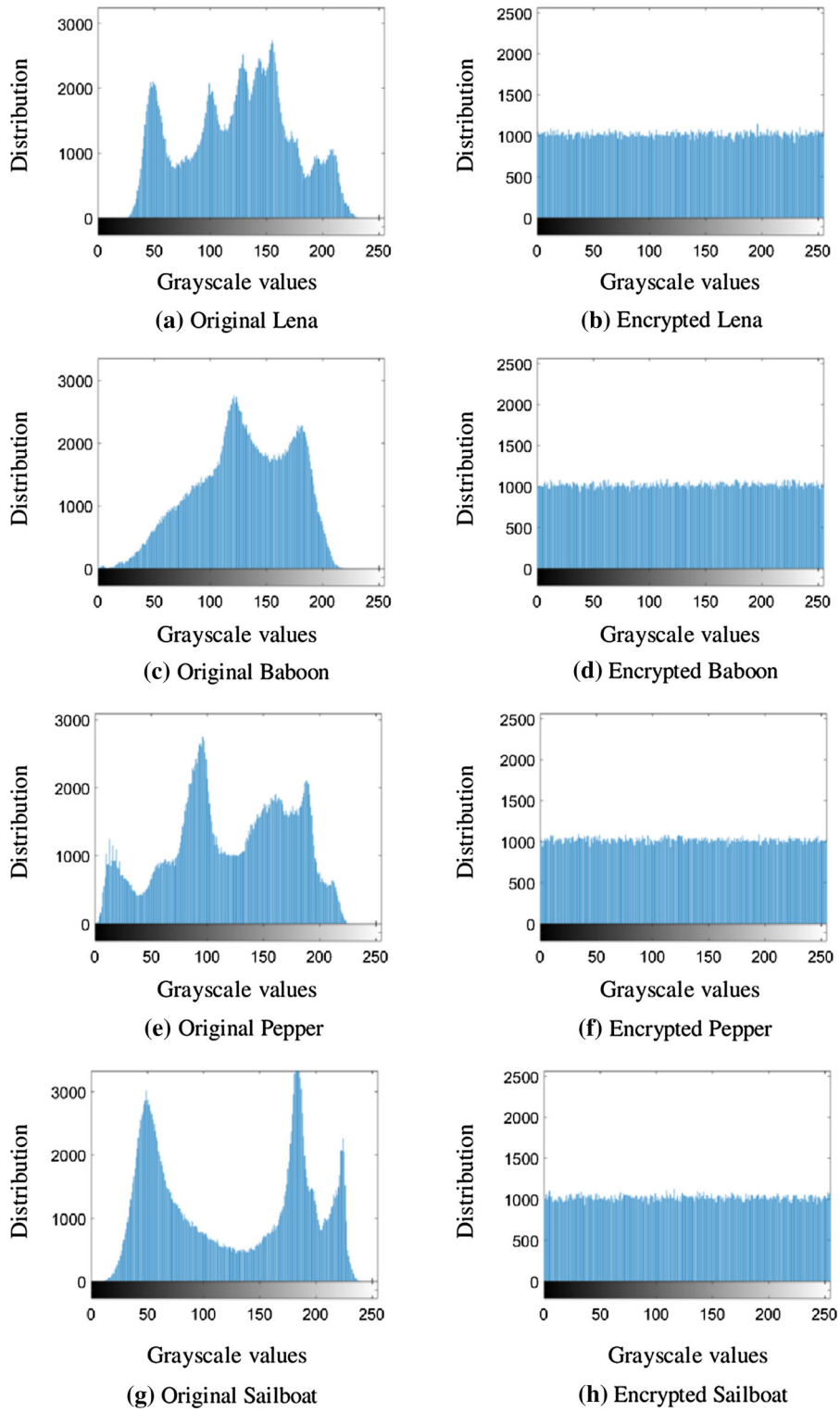


Fig. 5 Histogram outcomes of Group-1 images

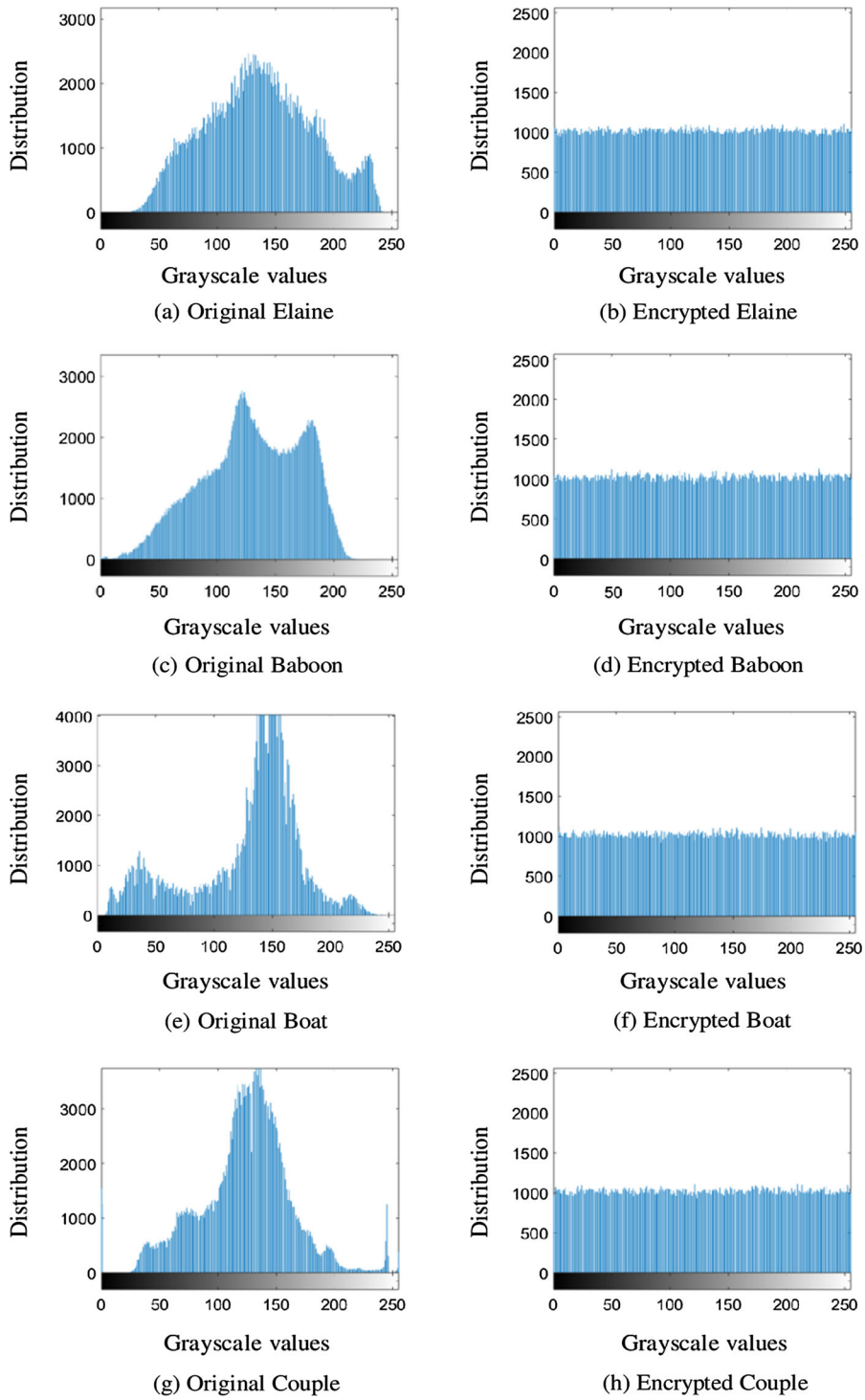


Fig. 6 Histogram outcomes of Group-2 images

Table 5 Histogram variance of the algorithm

Groups	Images	Original	Encrypted
Group-1	First image ("Lena")	633,400.0	1071.0
	Second image ("Baboon")	749,430.0	901.3750
	Third image ("Pepper")	780,660.0	886.8125
	Fourth image ("Sailboat")	719,620.0	1035.8
	Average	720,777.5	973.7469
Group-2	First image ("Elaine")	562,670.0	903.4609
	Second image ("Baboon")	749,430.0	927.2
	Third image ("Boat")	1,535,900.0	1045.6
	Fourth image ("Couple")	1,195,500.0	991.6
	Average	1,010,875.0	966.9652

Table 6 Comparison of Group-1 image average histogram variance results

Images	Proposed	Patro et al. [39]
First image ("Lena")	1071.0	982.5703
Second image ("Baboon")	901.3750	1088.7
Third image ("Pepper")	886.8125	1087.6
Fourth image ("Sailboat")	1035.8	1006.5
Average	973.7469	1041.34258

attack of brute force as opposed to the existing multiple image encryption methods [25–27, 29, 39].

4.2 Histogram analysis

Image histograms indicate how pixel values are distributed throughout the surface [45]. In an effective cipher image, the histogram must be uniform and substantially distinct from the original image, so that attackers will not be able to get any valuable data from the encrypted image [45, 46]. The histogram outputs of Group-1 and Group-2 images are shown in Figs. 5 and 6, respectively. It is obvious that the histogram of cipher images is distributed uniformly and very distinct from that of the actual images. This ensures that after encryption the redundancy of original images is effectively concealed and should not get any hint that statistical attacks can be applied.

4.3 Histogram variance analysis

The variances of the original and cipher image histograms are measured to determine the image pixel uniformity [47]. The images have greater pixel uniformity when the variances are smaller [47, 48]. It is measured by [47]:

$$\mathrm{var}(V_z) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{(v_i - v_j)^2}{2} \tag{6}$$

where $V_z = \{v_1, v_2, \ldots, v_{256}\}$, i and j denote the grayscale pixel values, v_i and v_j denote the number of pixels for each of the grayscale pixel values i and$j$, respectively. The variance of two image groups using

Table 7 Comparison of Group-2 image average histogram variance results

Images	Proposed	Zhang and Wang (second scheme) [26]	Tang et al. [25]	Patro et al. [39]
First image ("Elaine")	903.4609	1155.5	1047.375	993.7109
Second image ("Baboon")	927.2	989.6	1142.891	898.6406
Third image ("Boat")	1045.6	1111.6	1035.125	1044.9
Fourth image ("Couple")	991.6	929.6	1104.265	967.3047
Average	966.9652	1046.575	1082.414	976.13905

Table 8 Comparison of Chi-square test (χ^2_{test}) results

Groups	Images	Proposed	Patro et al. [39]	Testing results	
				$\chi^2_{255,0.05} = 293.2478$	$\chi^2_{255,0.01} = 310.457$
Group-1	First image ("Lena")	267.7480	245.6426	Pass	Pass
	Second image ("Baboon")	225.3438	272.1797	Pass	Pass
	Third image ("Pepper")	221.7031	271.9043	Pass	Pass
	Fourth image ("Sailboat")	258.9512	251.6367	Pass	Pass
	Average	243.43653	260.3408	Pass	Pass
Group-2	First image ("Elaine")	225.8652	248.4277	Pass	Pass
	Second image ("Baboon")	231.7910	224.6602	Pass	Pass
	Third image ("Boat")	261.4063	261.2148	Pass	Pass
	Fourth image ("Couple")	248.8965	241.8262	Pass	Pass
	Average	241.98975	244.0322	Pass	Pass

the suggested method is shown in Table 5. It is shown in the table that the cipher image variance is considerably reduced from the actual image variance. This implies the high grayscale uniformity of pixel values in cipher images. A comparative average variance of image Group-1 between the suggested method and the current Ref. [39] method is shown in Table 6. The results of the comparison show that the suggested method has less average variance than the method of Ref. [39], which indicates the high grayscale uniformity of the suggested scheme. A comparative average variance of image Group-2 between the suggested approach and the current Refs. [25, 26, 39] method is shown in Table 7. The results reveal that the suggested method has a lower average variance than the existing Refs. [25, 26, 39] algorithm, which implies the stronger uniformity of pixel grayscale values in the encrypted images of the suggested algorithm. Noting that both Group-1 and 2 images in Tables 6 and 7 are protected equally by the suggested method, the method in Ref. [39] does not apply equally to both Group-1 and 2 images. This means that the histogram variance of Ref. [39] in Group-1 image is higher, and Group-2 image is lower, but the histogram variance of the proposed algorithm in both Group-1 and 2 images is lower. Therefore, we can say our algorithm is more efficient than the others.

4.4 Chi-square test analysis

The uniformity in the histograms of encrypted images can also be justified through Chi-square test analysis [49, 50]. The low Chi-square value indicates high uniformity in encrypted image histograms [49, 50]. It is measured by

$$\chi^2_{\text{test}} = \sum_{k=0}^{255} \frac{(o_k - e_k)^2}{e_k} \tag{7}$$

where the observed frequency of k is denoted as o_k and the expected frequency of k is denoted as e_k. The expected frequency e_k is defined by:

$$e_k = \frac{M \times N}{256} \tag{8}$$

where the image size is $M \times N$. Table 8 presents the results of the Group-1 and Group-2 image Chi-square test analysis using the suggested method and also presents comparative results with the existing scheme [39]. Table 8 shows that in both the suggested scheme and the reference scheme [39], the hypothesis is accepted at both 5% and 1% levels of significance. This implies that the uniformity of the grayscale exists in the histograms of cipher images in both the suggested and Ref. [39] algorithms. It is also shown

Table 9 Adjacent pixel correlation results of the algorithm

Groups	Images	Original			Encrypted		
		D	*V*	*H*	*D*	*V*	*H*
Group-1	First image ("Lena")	0.9575	0.9846	0.9722	0.0007	− 0.0004	0.0011
	Second image ("Baboon")	0.7237	0.7620	0.8698	− 0.0017	− 0.0002	0.0003
	Third image ("Pepper")	0.9614	0.9796	0.9751	0.0002	− 0.0018	0.0014
	Fourth image ("Sailboat")	0.9584	0.9711	0.9745	− 0.0027	− 0.0004	0.0019
Group-2	First image ("Elaine")	0.9711	0.9736	0.9761	0.0019	0.0008	0.0010
	Second image ("Baboon")	0.7287	0.7511	0.8643	− 0.0002	− 0.0005	0.0012
	Third image ("Boat")	0.9261	0.9718	0.9385	− 0.0025	− 0.0012	0.0006
	Fourth image ("Couple")	0.8600	0.8914	0.9290	− 0.0004	0.0020	0.0009

Table 10 Comparison of Group-1 and Group-2 encrypted image adjacent pixel correlation results (10,000 pairs of pixels)

Groups	Images	Proposed			Patro et al. [39]		
		D	*V*	*H*	*D*	*V*	*H*
Group-1	First image ("Lena")	0.0007	− 0.0004	0.0011	− 0.0011	− 0.0010	0.0029
	Second image ("Baboon")	− 0.0017	− 0.0002	0.0003	0.0018	0.0019	− 0.0007
	Third image ("Pepper")	0.0002	− 0.0018	0.0014	0.0018	0.0024	− 0.0016
	Fourth image ("Sailboat")	− 0.0027	− 0.0004	0.0019	0.0019	0.0002	− 0.0030
Group-2	First image ("Elaine")	0.0019	0.0008	0.0010	0.0008	− 0.0046	− 0.0010
	Second image ("Baboon")	− 0.0002	− 0.0005	0.0012	− 0.0028	− 0.0047	0.0020
	Third image ("Boat")	− 0.0025	− 0.0012	0.0006	− 0.0046	− 0.0074	− 0.0079
	Fourth image ("Couple")	− 0.0004	0.0020	0.0009	− 0.0010	− 0.0029	− 0.0036

Table 11 Comparison of Group-2 encrypted image adjacent pixel correlation results (3000 pairs of pixels)

Images	Proposed			Tang et al. [25]		
	D	*V*	*H*	*D*	*V*	*H*
First image ("Elaine")	− 0.0037	0.0000	− 0.0029	0.0244	0.0199	− 0.0155
Second image ("Baboon")	− 0.0014	− 0.0011	0.0010	0.0026	0.0481	− 0.0486
Third image ("Boat")	− 0.0027	− 0.0020	0.0013	0.0126	0.0173	0.0460
Fourth image ("Couple")	− 0.0002	− 0.0018	0.0025	− 0.0242	0.0382	− 0.1015

in Table 8 that the proposed algorithm has a lower-average Chi-square value than Ref. [39] algorithm. Another finding in Table 8 indicates that the algorithm in Ref. [39] does not apply equally to both Group-1 and 2 images, while the suggested algorithm applies equally to both Group-1 and 2 images. This means the suggested method is more efficient than the method of Ref. [39].

4.5 Adjacent pixel correlation analysis

It measures the association of neighboring pixels in both the original and cipher images along diagonal (*D*), vertical (*V*), and horizontal (*H*) directions. In cipher images, there is usually a low correlation of adjacent pixels and a high correlation of adjacent pixels in original images [51, 52]. With strong

Table 12 Comparison of Group-2 encrypted image adjacent pixel correlation results (16,384 pairs of pixels)

Images	Proposed			Zhang and Wang (second scheme) [26]		
	D	*V*	*H*	*D*	*V*	*H*
First image ("Elaine")	0.0013	− 0.0001	0.0004	0.0043	− 0.0010	0.0003
Second image ("Baboon")	0.0003	− 0.0003	0.0010	0.0057	0.0021	− 0.0033
Third image ("Boat")	0.0017	0.0005	− 0.0005	0.0043	0.0020	0.0003
Fourth image ("Couple")	− 0.0007	− 0.0015	− 0.0006	0.0022	0.0038	− 0.0019

Table 13 Comparison of Group-1 encrypted image adjacent pixel correlation results

Algorithms	Images	Proposed		
		D	*V*	*H*
Proposed	First image ("Lena")	0.0006	− 0.0012	− 0.0005
	Second image ("Baboon")	− 0.0017	− 0.0007	− 0.0008
	Third image ("Pepper")	− 0.0016	− 0.0003	− 0.0005
	Fourth image ("Sailboat")	0.0011	0.0002	0.0024
Zhang and Wang (third scheme) in Ref. [29]	First image ("Lena")	0.0031	− 0.0066	− 0.0019
	Second image ("Baboon")	− 0.0001	− 0.0026	− 0.0012
	Third image ("Pepper")	0.0016	− 0.0019	− 0.0030
	Fourth image ("Sailboat")	0.0017	0.0012	− 0.0028
Zhang and Wang (second scheme) in Ref. [29]	First image ("Lena")	− 0.0020	− 0.0020	− 0.0113
	Second image ("Baboon")	− 0.0010	0.0014	− 0.0049
	Third image ("Pepper")	− 0.0052	0.0002	− 0.0071
	Fourth image ("Sailboat")	− 0.0020	− 0.0019	− 0.0049
Zhang and Wang (first scheme) in Ref. [29]	First image ("Lena")	0.8957	0.9476	0.9631
	Second image ("Baboon")	0.8955	0.9433	0.9592
	Third image ("Pepper")	0.8905	0.9414	0.9583
	Fourth image ("Sailboat")	0.9019	0.9473	0.9644
Tang et al. in Ref. [29]	First image ("Lena")	0.04720	0.0538	− 0.0673
	Second image ("Baboon")	0.0437	0.0505	− 0.0610
	Third image ("Pepper")	0.0385	0.0473	− 0.0515
	Fourth image ("Sailboat")	0.1133	0.1097	− 0.1326
Patro et al. [39]	First image ("Lena")	− 0.0024	0.0007	− 0.0015
	Second image ("Baboon")	0.0023	− 0.0009	0.0011
	Third image ("Pepper")	0.0009	0.0019	− 0.0013
	Fourth image ("Sailboat")	− 0.0014	− 0.0020	0.0005

correlation, the value of the correlation coefficient is much equal to + 1 or − 1 and the value of the correlation coefficient is much closer to 0 with low correlation. Expressions for calculating the correlation coefficient are as follows.

$$\text{corrcoff}_{st} = \frac{\text{covarr}(s,t)}{\sqrt{D(s)}\sqrt{D(t)}} \tag{9}$$

where

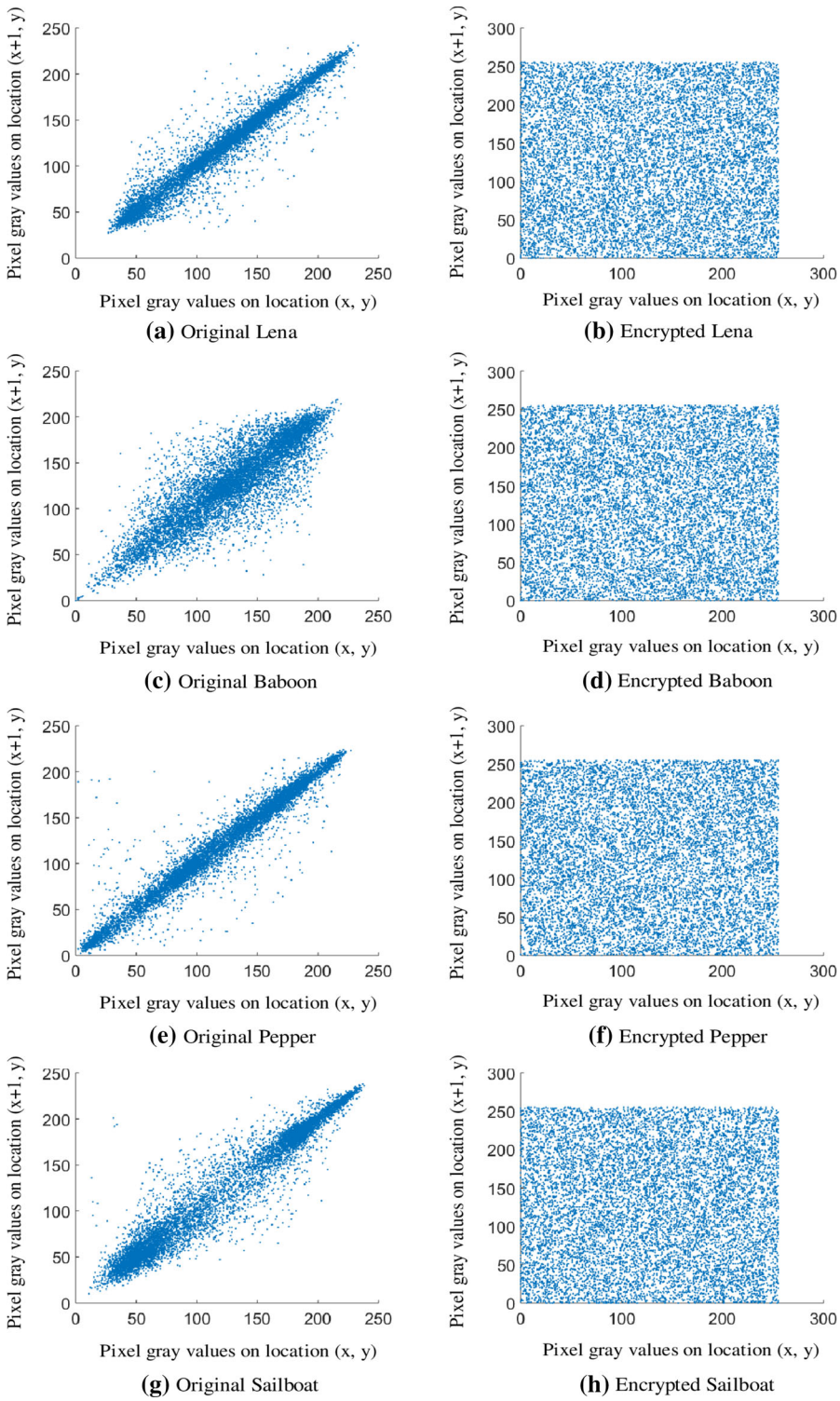


Fig. 7 Pixel correlation outputs of Group-1 images along the horizontal direction

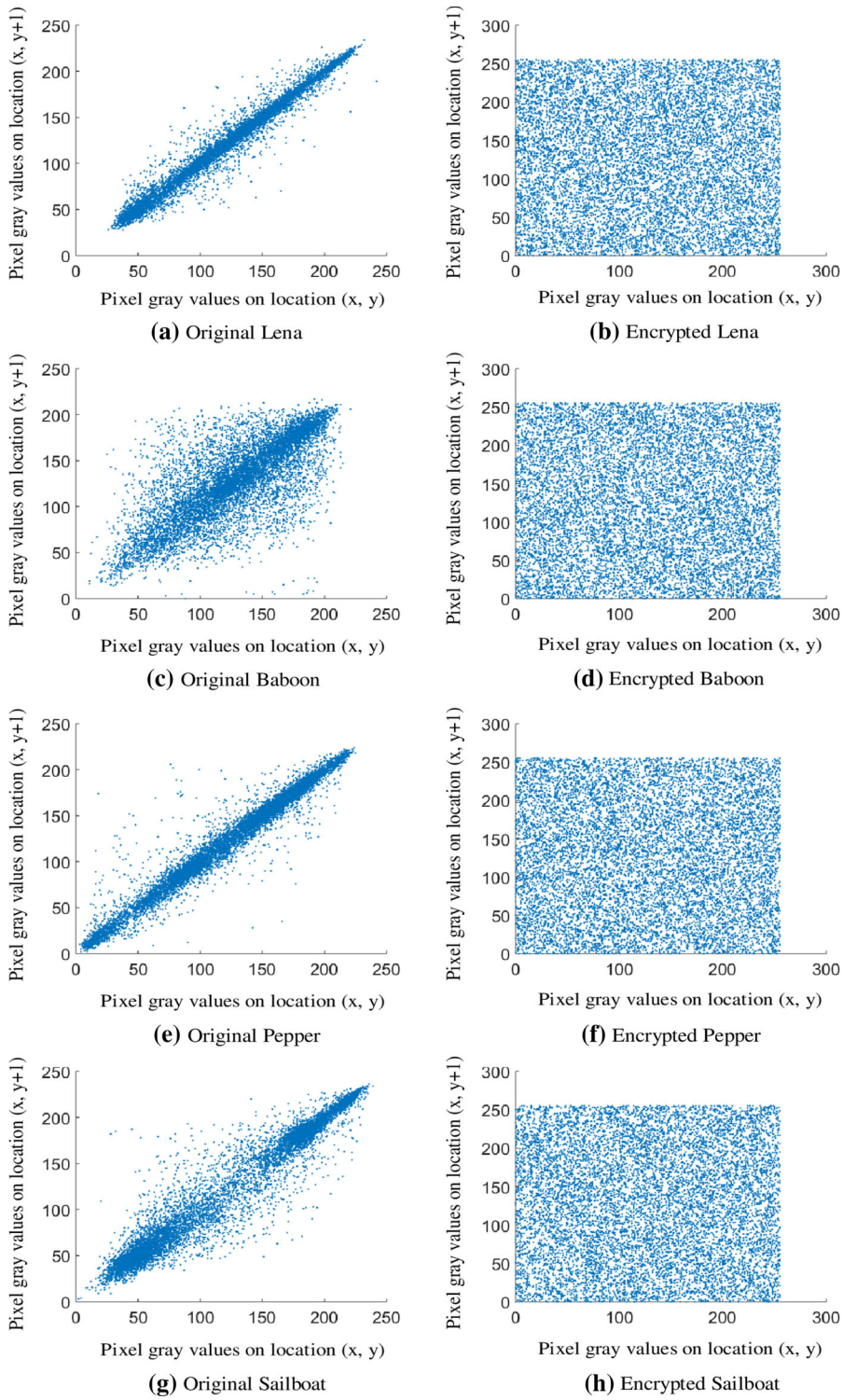


Fig. 8 Pixel correlation outputs of Group-1 images along the vertical direction

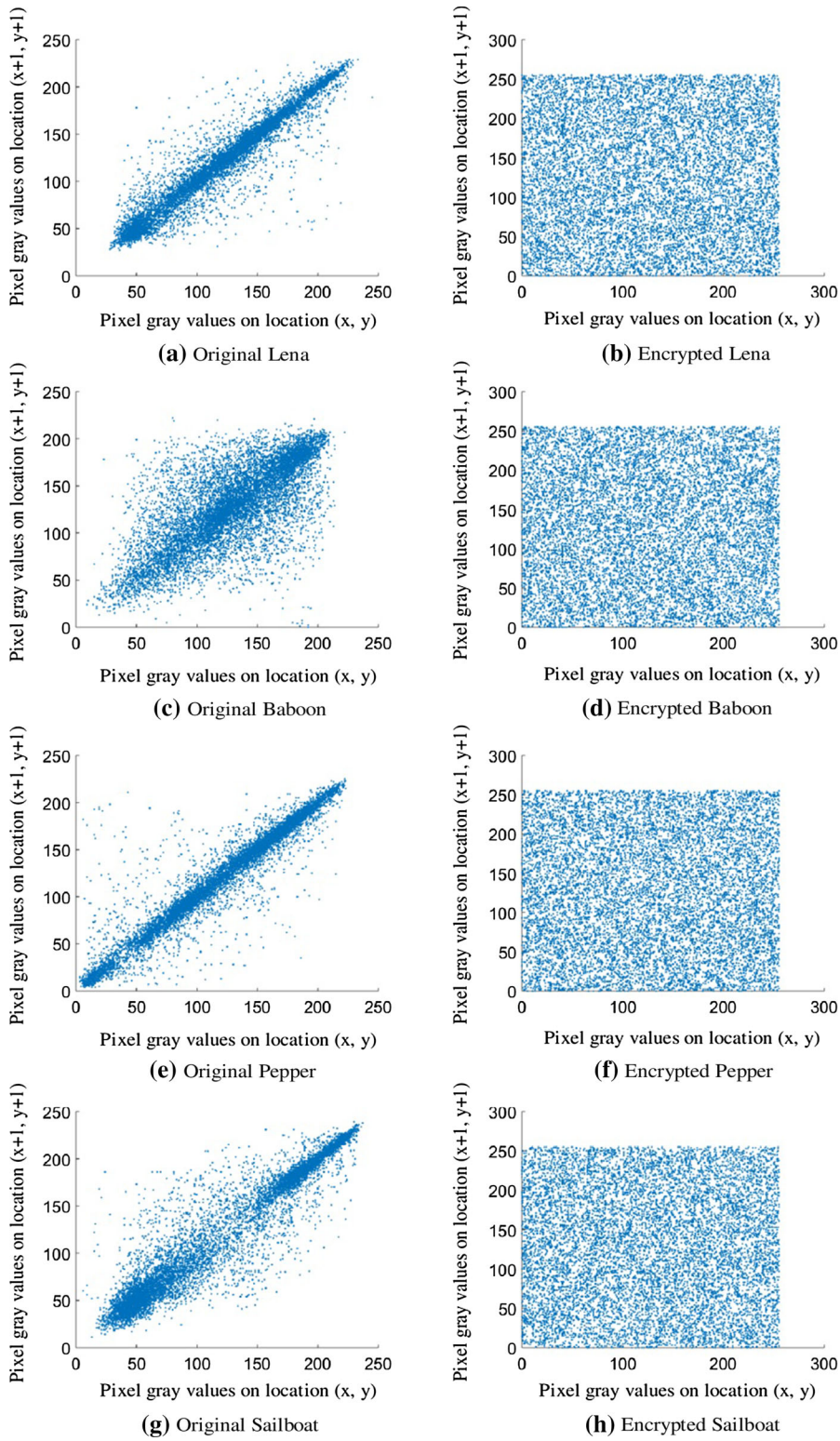


Fig. 9 Pixel correlation outputs of Group-1 images along the diagonal direction

$$\text{covarr}(s,t) = \frac{1}{M}\sum_{j=1}^{M}\left(s_j - E(s)\right)\left(t_j - E(t)\right) \tag{10}$$

$$D(s) = \frac{1}{M}\sum_{j=1}^{M}\left(s_j - E(s)\right)^2 \tag{11}$$

$$D(t) = \frac{1}{M}\sum_{j=1}^{M}\left(t_j - E(t)\right)^2 \tag{12}$$

$$E(s) = \frac{1}{M}\sum_{j=1}^{M} s_j \tag{13}$$

$$E(t) = \frac{1}{M}\sum_{j=1}^{M} t_j \tag{14}$$

In the above expressions (9)–(14), s and t are two neighboring pixel grayscale values, and M is adjacent pixel-pairs. Table 9 shows the results of the suggested method about the neighboring pixel correlation. The adjacent pixel association is performed on 10,000 pairs of adjacent pixels selected at random in the proposed algorithm. The results of Table 9 show that the correlation of neighboring pixels in the original images in horizontal, vertical, and diagonal directions is much closer to + 1 where as in cipher images; it is very close to 0. This indicates the suggested algorithm is highly resistant to the statistical attack. Table 10 presents a comparative encrypted image adjacent pixel correlation results between the proposed algorithm and Ref. [39] algorithm. Table 10 shows that the encrypted image correlation values of neighboring pixels in the suggested method are weaker than the method of Ref. [39]. This means that the suggested method has high statistical resistivity. Table 11 presents the results of encrypted images comparing the adjacent pixel association between the proposed algorithm and Ref. [25] algorithm. The comparison is carried out on 3000 pairs of neighboring pixels picked at random. From Table 11, the proposed algorithm indicates a weaker association of neighboring pixels in cipher images than Ref. [25] algorithm. It indicates the high resistivity of the proposed method to statistical attack than Ref. [25] algorithm. Table 12 provides a comparison of the results of adjacent pixel correlation between the suggested method and Ref. [26] method. The comparison is executed on 16,384 pairs of randomly chosen neighboring pixels. By analyzing the results of the comparison, it is found that the suggested method has a better value of the coefficient of correlation than Ref. [26] algorithm. Table 13 presents the comparison of adjacent pixel association results between the suggested method, Ref. [39] method, and the methods (Zhang and Wang's third, second, first method and Tang's method) analyzed in Ref. [29]. The results in Table 13 indicate the weaker association of neighboring pixels in the proposed method than the methods already being used. It shows the high statistical resistivity of the suggested method relative to Ref. [29]'s analyzed methods and Ref. [39]'s method. This reveals that the method suggested is effective compared to the current methods.

Figures 7, 8 and 9 show the Group-1 image correlation distribution of neighboring pixels. In these figures (Figs. 7, 8 and 9), it is shown that the

Table 14 Comparison of Group-1 and 2 image MSE results

Groups	Images	Proposed		Patro et al. [39]	
		E versus *O*	*D* versus *O*	*E* versus *O*	*D* versus *O*
Group-1	First image ("Lena")	7764.3	0	7762.6	0
	Second image ("Baboon")	7265.4	0	7257.3	0
	Third image ("Pepper")	8477.3	0	8465.8	0
	Fourth image ("Sailboat")	9742.7	0	9730.3	0
	Average	8312.425	0	8304.0	0
Group-2	First image ("Elaine")	7656.1	0	7648.5	0
	Second image ("Baboon")	8266.3	0	7263.6	0
	Third image ("Boat")	8659.2	0	7640.5	0
	Fourth image ("Couple")	7099.8	0	7094.2	0
	Average	7920.35	0	7411.7	0

Table 15 Comparison of Group-1 and 2 image PSNR results

Groups	Images	Proposed		Patro et al. [39]	
		E versus *O*	*D* versus *O*	*E* versus *O*	*D* versus *O*
Group-1	First image ("Lena")	9.2301	∞	9.2307	∞
	Second image ("Baboon")	9.5212	∞	9.5231	∞
	Third image ("Pepper")	8.8442	∞	8.8541	∞
	Fourth image ("Sailboat")	8.2440	∞	8.2495	∞
	Average	8.959875	∞	8.96435	∞
Group-2	First image ("Elaine")	8.2914	∞	9.2950	∞
	Second image ("Baboon")	9.5136	∞	9.5193	∞
	Third image ("Boat")	9.1117	∞	9.2996	∞
	Fourth image ("Couple")	8.6120	∞	9.6217	∞
	Average	8.882175	∞	9.4339	∞

neighboring pixels are linearly correlated in the actual images, indicating a strong neighboring pixel association, whereas the neighboring pixels are distributed in encrypted images over the entire surface, indicating low association of neighboring pixels. This indicates the high statistical attack resistivity using the suggested method.

4.6 Mean-square error (MSE) and peak signal-to-noise ratio (PSNR) analysis

MSE tests the difference among input and encrypted images, as well as between input and decrypted images. MSE's high value reveals the big difference between the actual and cipher images [53, 54]. The MSE among the input image and its decryption is zero [53]. MSE is defined as:

$$\mathrm{MSE}_{\mathrm{OE}} = \frac{1}{M_g \times N_g} \sum_{i=1}^{M_g} \sum_{j=1}^{N_g} \left(O_{ij} - E_{ij}\right)^2 \tag{15}$$

$$\mathrm{MSE}_{\mathrm{OD}} = \frac{1}{M_g \times N_g} \sum_{i=1}^{M_g} \sum_{j=1}^{N_g} \left(O_{ij} - D_{ij}\right)^2 \tag{16}$$

where '*O*' is the "input image", '*E*' is the "encrypted image", '*D*' is the decrypted image, $\mathrm{MSE}_{\mathrm{OE}}$ is the "MSE between input and encrypted images", $\mathrm{MSE}_{\mathrm{OD}}$ is the "MSE between input and decrypted images".

The comparative MSE results among the suggested method and Ref. [39] method are provided in Table 14. It is found in the results that the suggested method has larger MSE value than Ref. [39] algorithm. This shows that there is a substantial difference between input and cipher images in the proposed algorithm compared to Ref. [39] algorithm. Table 14 also shows the MSE zero value between the input and decrypted images. This

Table 16 Comparison of Group-1 and Group-2 image information entropy results

Groups	Images	Proposed		Patro et al. [39]
		Original	Encrypted	Encrypted
Group-1	First image ("Lena")	7.4451	7.9994	7.9994
	Second image ("Baboon")	7.3583	7.9994	7.9993
	Third image ("Pepper")	7.5937	7.9994	7.9993
	Fourth image ("Sailboat")	7.4842	7.9993	7.9993
	Average	7.470325	7.9994	7.9993
Group-2	First image ("Elaine")	7.5060	7.9994	7.9993
	Second image ("Baboon")	7.3583	7.9993	7.9994
	Third image ("Boat")	7.1914	7.9994	7.9993
	Fourth image ("Couple")	7.2010	7.9994	7.9993
	Average	7.314175	7.9994	7.9993

Table 17 Comparison of Group-2 encrypted image information entropy results

Images	Proposed	Zhang and Wang (2nd scheme) [26]	Tang et al. [25]
First image ("Elaine")	7.9994	7.9992	7.9993
Second image ("Baboon")	7.9993	7.9993	7.9992
Third image ("Boat")	7.9994	7.9992	7.9993
Fourth image ("Couple")	7.9994	7.9994	7.9992
Average	7.9994	7.9993	7.9992

Table 18 Comparison of Group-1 encrypted image information entropy results

Images	Proposed	Zhang and Wang (3rd scheme) in Ref. [29]	Zhang and Wang (1st scheme) in Ref. [29]	Tang et al. in Ref. [29]
First image ("Lena")	7.9994	7.9993	7.6169	7.9991
Second image ("Baboon")	7.9994	7.9993	7.6015	7.9994
Third image ("Pepper")	7.9994	7.9993	7.6138	7.9993
Fourth image ("Sailboat")	7.9993	7.9994	7.6269	7.9938
Average	7.9994	7.9993	7.6148	7.9979

Table 19 Comparison of information entropy between multiple image and currently developed single-image encryption methods

Encryption methods	Information entropy
Proposed (average of Group-1 images)	7.9994
Proposed (average of Group-2 images)	7.9994
Tsafack et al. [56] (average of grayscale images: "GImg01", "GImg02", "GImg03")	7.99932
Tsafack et al. [56] (Average of color images: "CImg01", "CImg02", "CImg03")	7.99976
Abd El-Latif et al. [57] (average of Frame # 1 of each frame sequence color images)	7.99838
Sambas et al. [58] (average of color images: "Baboon", "Airplane", "Sailboat", "Peppers", "House", "Splash")	7.99976

indicates similarity between the input and decrypted images.

PSNR tests the quality estimates of the encrypted image against the original image. A small PSNR value reveals significant variations between the input and the cipher images [53, 54]. The PSNR among the input image and its decryption is infinite [53]. It is defined by:

$$\mathrm{PSNR}_{\mathrm{OE}} = 20\log_{10}\left[\frac{I_{\max}}{\sqrt{\mathrm{MSE}_{\mathrm{OE}}}}\right] \tag{17}$$

$$\mathrm{PSNR}_{\mathrm{OD}} = 20\log_{10}\left[\frac{I_{\max}}{\sqrt{\mathrm{MSE}_{\mathrm{OD}}}}\right] \tag{18}$$

where $I_{\max}$ is the highest pixel value available for the image, $\mathrm{PSNR}_{\mathrm{OE}}$ is the "PSNR between input and encrypted images", $\mathrm{PSNR}_{\mathrm{OD}}$ is the "PSNR between input and decrypted images".

Table 15 presents a comparative PSNR results among the suggested method and Ref. [39] method. Like MSE comparison in Table 14, the PSNR comparison is executed on two image groups in Table 15. Table 15 results indicate that the suggested method has a lower PSNR value than the method in Ref. [39]. It indicates that there is a considerable difference between input and cipher images in the proposed algorithm compared to Ref. [39] algorithm.

Table 15 also shows the PSNR infinite value between the input and decrypted images. This indicates that the input and decrypted images are similar.

4.7 Information entropy analysis

It is a method to calculate the degree of pixel randomness in encrypted images [55]. The higher the pixel randomness, the greater the image entropy. The higher the entropy, the greater the information security. The ideal entropy value for an image of 256-grayscale is 8. The nearer it is to 8, the more it prevents information leakage. The entropy is calculated as,

$$H = -\sum_{j=0}^{255} P_r(j) \log_2 P_r(j) \tag{19}$$

where H is the entropy of an image and $P_r(j)$ is the probability of the symbol j. Table 16 provides a comparative entropy results among the suggested method and Ref. [39] method. It is noted in the comparison table that the suggested method has a higher entropy value than Ref. [39] method. Table 17 provides a table for comparison among the suggested method and the method Refs. [25, 26]. In the table, the higher entropy value of the suggested method is also found. The comparative entropy results among the suggested method and the methods analyzed in Ref. [29] (Zhang and Wang's third and first method, and Tang et al.'s method) are shown in Table 18. The comparison table indicates that the suggested method has an entropy value larger than the methods analyzed in Ref. [29]. Comparison Tables 16, 17 and 18 reveal that the proposed algorithm strongly avoids information leakage as opposed to the others. Hence, the proposed algorithm is highly effective compared to existing algorithms for multiple image encryption.

A comparison of information entropy between the proposed method of multiple image encryption and some recently developed methods of single image encryption is provided in Table 19. It is found in Table 19 that all the methods of multiple-image and single-image encryption efficiently resist the entropy attack. However, the method proposed avoids the entropy attack better than Ref. [56] (grayscale images) and Ref. [57] (color images) methods. On the other hand, the method proposed resists the entropy attack marginally less than the methods in Ref. [56] (color images) and Ref. [58] (color images).

4.8 Differential attack analysis

It measures the sensitivity of the ciphertext image toward the plaintext image. The more sensitivity the ciphertext has toward the plaintext, the greater the algorithm's resistivity against the differential attacks [10, 59, 60]. Two widely used security measures for measuring differential attack analysis are Numbers of Pixel Changing Rate (NPCR) and UACI (Unified Average Changing Intensity) [10, 59, 60].

NPCR measures the rate of change of pixels in an encrypted image. Equations for measuring the NPCR are,

Table 20 UACI and NPCR results of the suggested method

Groups	Images	UACI (%)			NPCR (%)		
		Min.	Max.	Avg.	Min.	Max.	Avg.
Group-1	First image ("Lena")	33.3857	33.5866	33.4865	99.5915	99.6672	99.6280
	Second image ("Baboon")	33.3740	33.5795	33.4797	99.5873	99.6649	99.6236
	Third image ("Pepper")	33.3737	33.5737	33.4772	99.5816	99.6608	99.6200
	Fourth image ("Sailboat")	33.3696	33.5790	33.4801	99.5847	99.6590	99.6198
Group-2	First image ("Elaine")	33.3735	33.5741	33.4754	99.5889	99.6557	99.6187
	Second image ("Baboon")	33.3504	33.5675	33.4726	99.5838	99.6468	99.6132
	Third image ("Boat")	33.3657	33.5725	33.4745	99.5785	99.6544	99.6191
	Fourth image ("Couple")	33.3028	33.5803	33.4874	99.5876	99.6533	99.6175

Table 21 Comparison of Group-1 and Group-2 image UACI and NPCR results

Groups	Images	Average UACI		Average NPCR	
		Proposed	Patro et al. [39]	Proposed	Patro et al. [39]
Group-1	First image ("Lena")	33.4865	33.4853	99.6280	99.6223
	Second image ("Baboon")	33.4797	33.4789	99.6236	99.6196
	Third image ("Pepper")	33.4772	33.4793	99.6200	99.6280
	Fourth image ("Sailboat")	33.4801	33.4799	99.6198	99.6178
	Average	33.480875	33.48085	99.62285	99.621925
Group-2	First image ("Elaine")	33.4754	33.4751	99.6187	99.6185
	Second image ("Baboon")	33.4726	33.4703	99.6132	99.6113
	Third image ("Boat")	33.4745	33.4723	99.6191	99.6182
	Fourth image ("Couple")	33.4874	33.4910	99.6175	99.6188
	Average	33.477475	33.477175	99.617125	99.6167

Table 22 Comparison of Group-1 image UACI and NPCR results

Algorithms	Images	UACI	NPCR
Proposed	First image ("Lena")	33.4865	99.6280
	Second image ("Baboon")	33.4797	99.6236
	Third image ("Pepper")	33.4772	99.6200
	Fourth image ("Sailboat")	33.4801	99.6198
	Average	33.480875	99.62285
Zhang and Wang (3rd scheme) in Ref. [29]	First image ("Lena")	33.45	99.61
	Second image ("Baboon")	33.43	99.62
	Third image ("Pepper")	33.46	99.63
	Fourth image ("Sailboat")	33.41	99.63
	Average	33.4375	99.6225
Zhang and Wang (first scheme) in Ref. [29]	First image ("Lena")	0	0
	Second image ("Baboon")	0	0
	Third image ("Pepper")	0	0
	Fourth image ("Sailboat")	0	0
	Average	0	0
Tang et al. in Ref. [29]	First image ("Lena")	0	0
	Second image ("Baboon")	0	0
	Third image ("Pepper")	0	0
	Fourth image ("Sailboat")	0	0
	Average	0	0

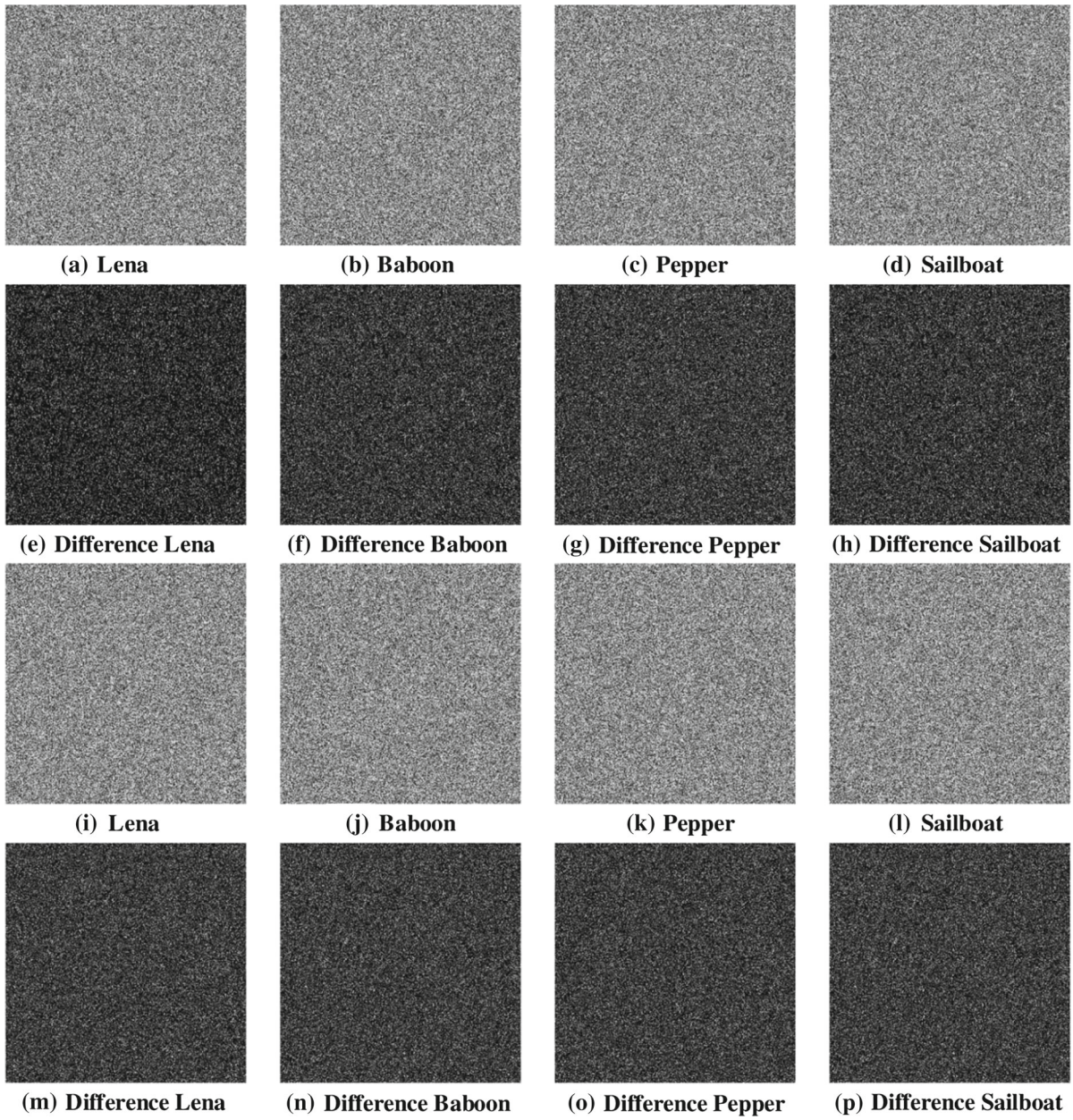


Fig. 10 Key sensitivity outcomes of Group-1 images: **a–d** Cipher images by changed key *km*; **e–h** Figs. 2(e)–9(a), Figs. 2(f)–9(b), Figs. 2(g)–9(c), Figs. 2(h)–9(d); **i–l** Cipher images by changed key *mm*; **m–p** Figs. 2(e)–9(i), Figs. 2(f)–9(j), Figs. 2(g)–9(k), Figs. 2(h)–9(l)

$$\mathrm{NPCR}(C_1, C_2) = \frac{\sum_{i,j} D_{C_1,C_2}(i,j)}{M_g \times N_g} \times 100\% \tag{20}$$

where $M_g \times N_g$ is the image size and $D_{C_1,C_2}(i,j)$ is defined by,

$$D_{C_1,C_2}(i,j) = \begin{cases} 1 & if\ C_1(i,j) \neq C_2(i,j) \\ 0 & if\ C_1(i,j) = C_2(i,j) \end{cases} \tag{21}$$

where C_1 is the actual cipher image and C_2 is the modified cipher image. The NPCR is $99.6094 \approx 99.61\%$ in an ideal case [10, 60].

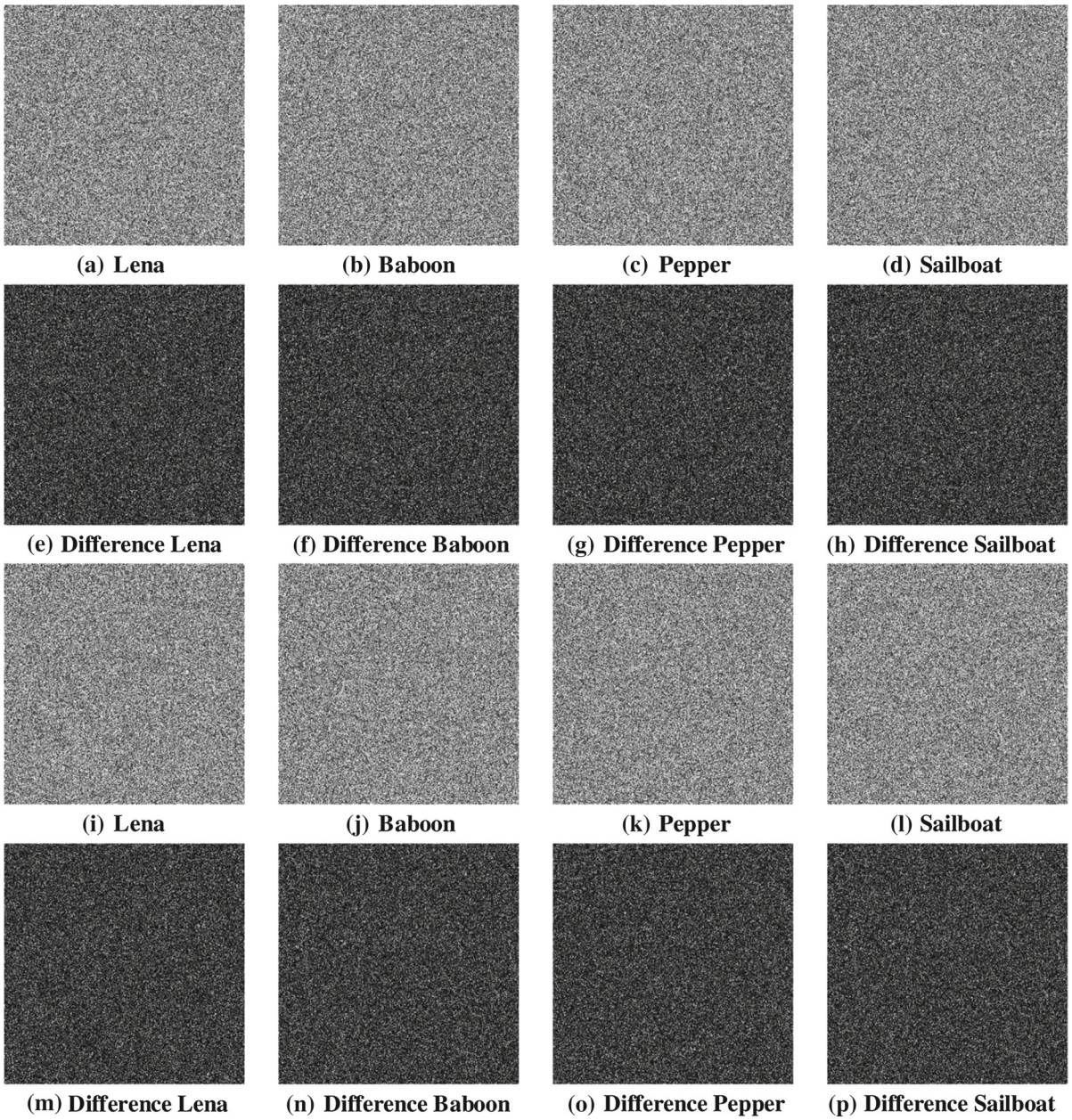


Fig. 11 Key sensitivity outcomes of Group-1 images: **a–d** Cipher images by changed key *kn*; **e–h** Figs. 2(e)–10(a), Figs. 2(f)–10(b), Figs. 2(g)–10(c), Figs. 2(h)–10(d); **i–l** Cipher images by changed key *nn*; **m–p** Figs. 2(e)–10(i), Figs. 2(f)–10(j), Figs. 2(g)–10(k), Figs. 2(h)–10(l)

UACI determines the relative intensity varying between the original and cipher images. Equation to measure the UACI is

$$\mathrm{UACI}(C_1, C_2) = \frac{\sum_{i,j} |C_1(i,j) - C_2(i,j)|}{M_g \times N_g \times 255} \times 100\% \tag{22}$$

The ideal value of UACI is $33.4635 \approx 33.46\%$ [10, 60]. Table 20 presents the UACI and NPCR results of the suggested algorithm. It is seen in the table that all the images' average UACI and NPCR results are greater than their ideal values. This shows that the suggested method strongly protects the

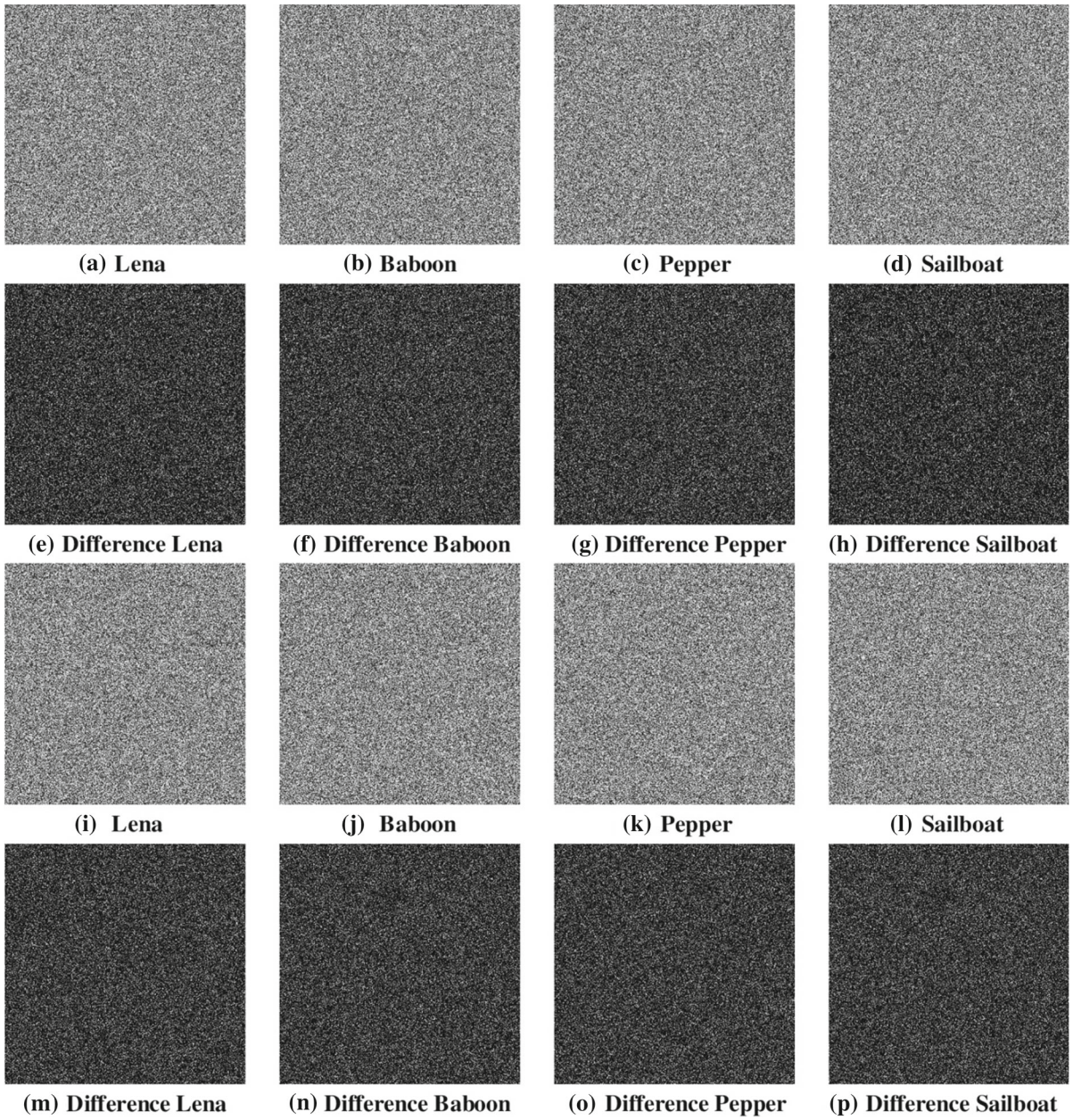


Fig. 12 Key sensitivity outcomes of Group-1 images: **a–d** Cipher images by changed key *kp*; **e–h** Figs. 2(e)–11(a), Figs. 2(f)–11(b), Figs. 2(g)–11(c), Figs. 2(h)–11(d); **i–l** Cipher images by changed key *pp*; **m–p** Figs. 2(e)–11(i), Figs. 2(f)–11(j), Figs. 2(g)–11(k), Figs. 2(h)–11(l)

differential attack. The comparative UACI and NPCR results among the suggested method and Ref. [39] method are provided in Table 21. By analyzing the results of the comparison, it is found that the method suggested has a better-average UACI and NPCR than the method Ref. [39]. Table 22 provides the comparative UACI and NPCR among the suggested method and the methods analyzed in Ref. [29]. By analyzing the results of the comparison, it is found that the suggested method has better-average results for UACI and NPCR than the methods evaluated in Ref. [29]. This shows that the suggested method is efficient in comparison with the other multiple image encryption methods referred.

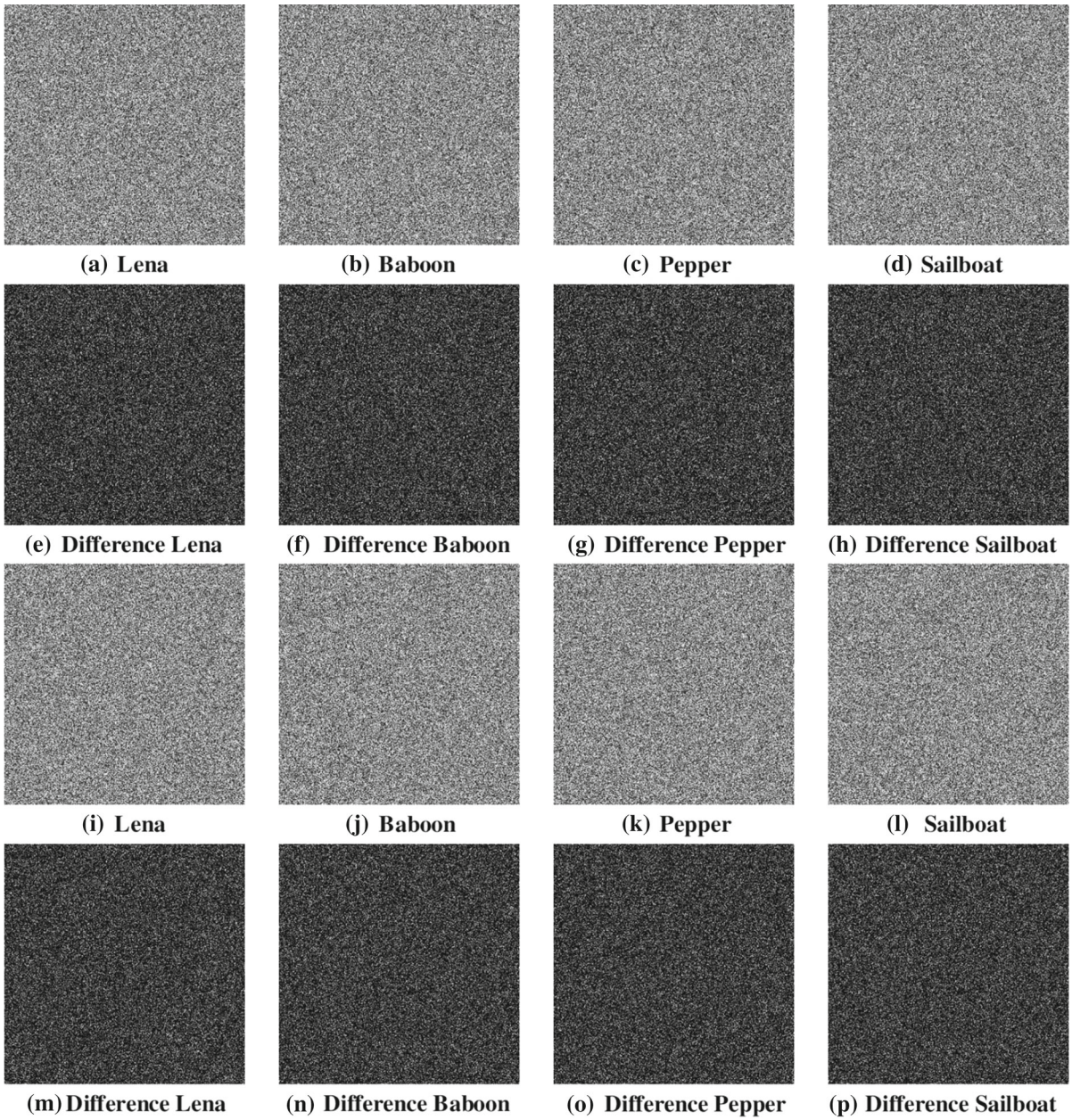


Fig. 13 Key sensitivity outcomes of Group-1 images: **a–d** Cipher images by changed key *kq*; **e–h** Figs. 2(e)–12(a), Figs. 2(f)–12(b), Figs. 2(g)–12(c), Figs. 2(h)–12(d); **(i-l)** Cipher images by changed key *qq*; **m–p** Figs. 2(e)–12(i), Figs. 2(f)–12(j), Figs. 2(g)–12(k), Figs. 2(h)–12(l)

4.9 Key sensitivity analysis

Eight number of PWLCM system-based keys are used in the suggested scheme. To prevent the scheme against the brute-force attack, all keys should be highly sensitive. The key sensitivity is accomplished by changing one of the eight keys and then the encryption operation. Finally, the rate of change of pixel values in modified cipher images relative to the original cipher images is observed. Figures 10, 11, 12 and 13 show the key sensitivity outcomes of the Group-1 images. By analyzing the two cipher image differences in Figs. 10, 11, 12 and 13, it is found that a dramatic change in the cipher images occurs by

Table 23 Group-1 image key sensitivity results

Images	$km + 10^{-15}$		$mm + 10^{-15}$		$kn + 10^{-15}$	
	UACI (%)	NPCR (%)	UACI (%)	NPCR (%)	UACI (%)	NPCR (%)
First image ("Lena")	33.4558	99.6131	33.4196	99.5865	33.5727	99.6216
Second image ("Baboon")	33.4835	99.6176	33.5628	99.6075	33.5133	99.6086
Third image ("Pepper")	33.4535	99.6038	33.4501	99.6178	33.3852	99.6090
Fourth image ("Sailboat")	33.7545	99.6197	33.4832	99.6140	33.4375	99.6117
Average	33.5368	99.6136	33.4789	99.6065	33.4772	99.6127
Images	$nn + 10^{-15}$		$kp + 10^{-15}$		$pp + 10^{-15}$	
	UACI (%)	NPCR (%)	UACI (%)	NP CR(%)	UACI (%)	NPCR (%)
First image ("Lena")	33.4075	99.6082	33.4540	99.6106	33.4011	99.6017
Second image ("Baboon")	33.5100	99.5918	33.4440	99.6003	33.4812	99.6166
Third image ("Pepper")	33.5015	99.6037	33.3437	99.6098	33.5042	99.5987
Fourth image ("Sailboat")	33.4704	99.6193	33.3226	99.6090	33.3917	99.6185
Average	33.4724	99.6058	33.3911	99.6074	33.4446	99.6089

Images	$kq + 10^{-15}$		$qq + 10^{-15}$	
	UACI (%)	NPCR (%)	UACI (%)	NPCR (%)
First image ("Lena")	33.5352	99.6147	33.4586	99.6014
Second image ("Baboon")	33.4619	99.6105	33.4782	99.6143
Third image ("Pepper")	33.4683	99.5949	33.4505	99.6109
Fourth image ("Sailboat")	33.4250	99.5930	33.4593	99.5979
Average	33.4726	99.6033	33.4617	99.6061

modifying only one key out of eight keys. This shows the algorithm being proposed is highly key sensitive. Tables 23 and 24 present the key sensitivity results in the form of UACI and NPCR. If the NPCR value exceeds 99% and if the UACI value exceeds 33%, then the keys are highly sensitive to the algorithm. The UACI and NPCR values of all images are found in Tables 23 and 24 to be greater than 33% and 99%, respectively. This shows the method proposed is highly sensitive to all keys.

4.10 Noise attack analysis

The robustness of a cryptosystem toward noise is one of the most significant challenges in real-world communication technology [61]. The suggested cryptosystem is robust against noise. In this cryptosystem, two kinds of noise are used to illustrate the effectiveness of the algorithm, such as Gaussian noise and salt and pepper noise. Both noise analyses are carried out by adding some noise to the encrypted image and subsequently retrieving the decrypted image. Finally, the disparity between the original image and the decrypted images recovered indicates the algorithm's efficacy toward noise. The difference between the original image and the decrypted image retrieved is calculated by NPCR and UACI. The lower value of NPCR and UACI shows the better resistivity towards the noise. The Gaussian noise attack analysis results of the Group-1 and Group-2 images are shown in Figs. 14 and 15, respectively. The robustness of Group-1 images against Gaussian noise is seen in Figs. 14 and 15 at Mean = 0 and Variance = 0.0001, 0.0003, 0.0005. Tables 25 and 26, respectively, show the corresponding NPCR and UACI results of the Group-1 and Group-2 images. Table 27 shows the comparison of the results of the Gaussian noise attack analysis. In Table 27, it is found that the proposed

algorithm has lower value of NPCR and UACI than the algorithms in Ref. [47, 50, 61]. This shows that the proposed algorithm resists the Gaussian noise attack better than Ref. [47, 50, 61] algorithms.

Figures 16 and 17, respectively, show the outcomes of the salt and pepper noise attack analysis of the Group-1 and Group-2 images. In these figures, it is observed that the proposed algorithm strongly resists the salt and pepper noise attack. The NPCR and UACI results of the Group-1 and Group-2 images, respectively, are shown in Tables 28 and 29. The comparison of salt and pepper noise attack analysis results is shown in Table 30. It is noticed in Table 30 that the proposed algorithm has a lower NPCR and UACI value than the algorithm in Ref. [50]. This illustrates that the salt and pepper noise attack is also resisted by the proposed algorithm.

Fig. 14 Gaussian noise attack analysis results of Group-1 ▶ images: At mean = 0 and variance = 0.0001 **a–d** encrypted images, **e–h** decrypted images; At Mean = 0 and Variance = 0.0003 **i–l** encrypted images, **m–p** decrypted images; At Mean = 0 and Variance = 0.0005 **q–t** encrypted images, **u–x** decrypted images

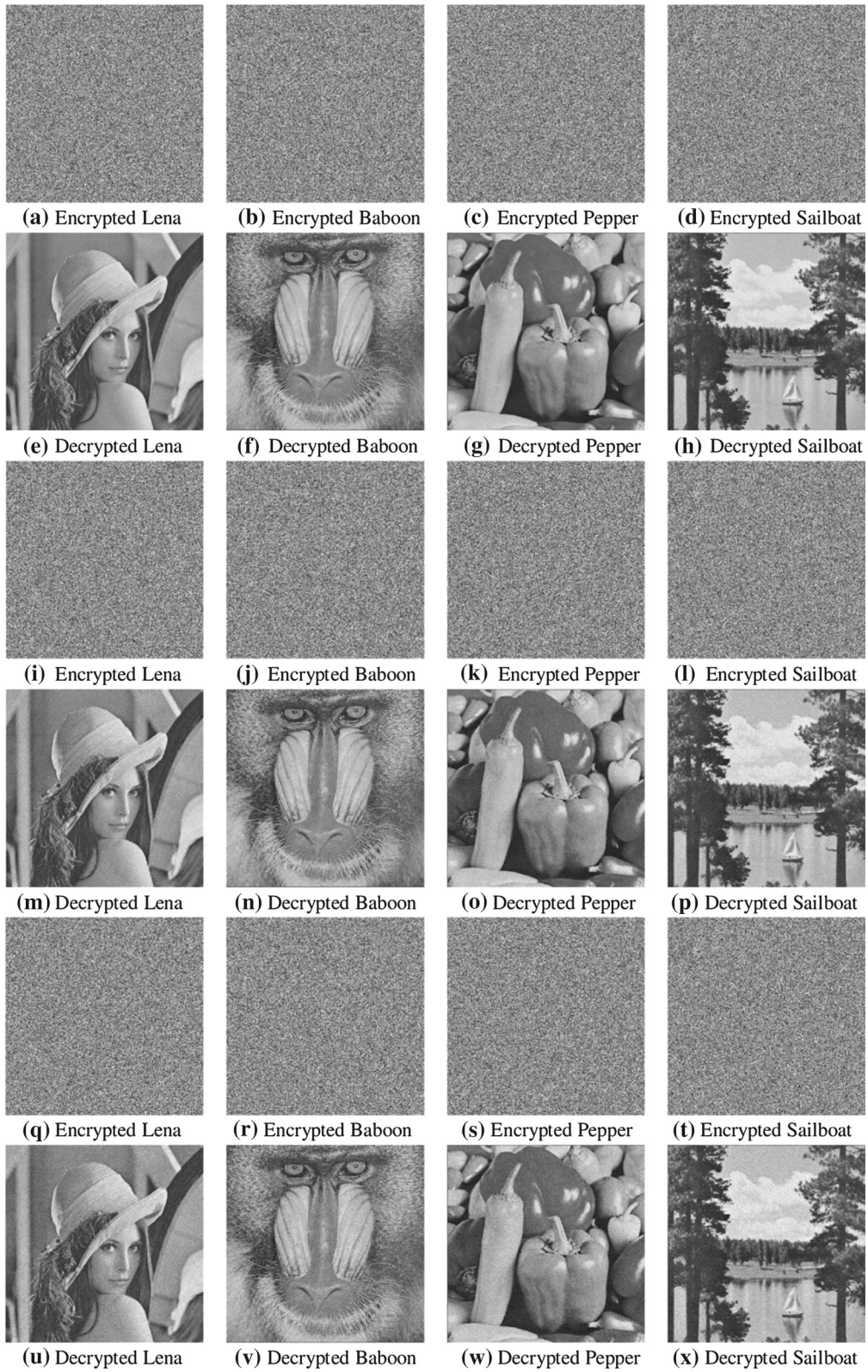
4.11 Cryptanalysis

The two attacks commonly used by attackers to target cryptographic algorithms are the chosen-plaintext attack and the chosen-ciphertext attack [62–64]. The chosen-plaintext and/or chosen-ciphertext attack with all-one or all-zero images has breached several image encryption techniques [64–66]. In the proposed algorithm, both the attacks are implemented.

Table 24 Group-2 image key sensitivity results

Images	$km + 10^{-15}$		$mm + 10^{-15}$		$kn + 10^{-15}$	
	UACI (%)	NPCR (%)	UACI (%)	NPCR (%)	UACI (%)	NPCR (%)
First image (“Elaine”)	33.4588	99.6115	33.4543	99.6197	33.3927	99.6094
Second image (“Baboon”)	33.4181	99.6002	33.4683	99.6101	33.4717	99.6262
Third image (“Boat”)	33.4348	99.6013	33.4475	99.5995	33.4954	99.6098
Fourth image (“Couple”)	33.4640	99.6177	33.3965	99.6098	33.4216	99.6216
Average	33.4439	99.6077	33.4417	99.6098	33.4454	99.6168
Images	$nn + 10^{-15}$		$kp + 10^{-15}$		$pp + 10^{-15}$	
	UACI (%)	NPCR (%)	UACI (%)	NPCR (%)	UACI (%)	NPCR (%)
First image (“Elaine”)	33.4296	99.6159	33.4616	99.6144	33.5048	99.6143
Second image (“Baboon”)	33.3886	99.6155	33.4625	99.6072	33.4637	99.5888
Third image (“Boat”)	33.4877	99.5991	33.4781	99.6063	33.4544	99.6250
Fourth image (“Couple”)	33.4473	99.6265	33.4388	99.6059	33.4359	99.6082
Average	33.4383	99.6143	33.4603	99.6085	33.4647	99.6091

Images	$kq + 10^{-15}$		$qq + 10^{-15}$	
	UACI (%)	NPCR (%)	UACI (%)	NPCR (%)
First image (“Elaine”)	33.4769	99.6059	33.4754	99.5953
Second image (“Baboon”)	33.4496	99.5941	33.4578	99.6151
Third image (“Boat”)	33.4446	99.6002	33.3949	99.5930
Fourth image (“Couple”)	33.4218	99.5914	33.5043	99.6185
Average	33.4482	99.5979	33.4581	99.6055



(a) Encrypted Lena (b) Encrypted Baboon (c) Encrypted Pepper (d) Encrypted Sailboat

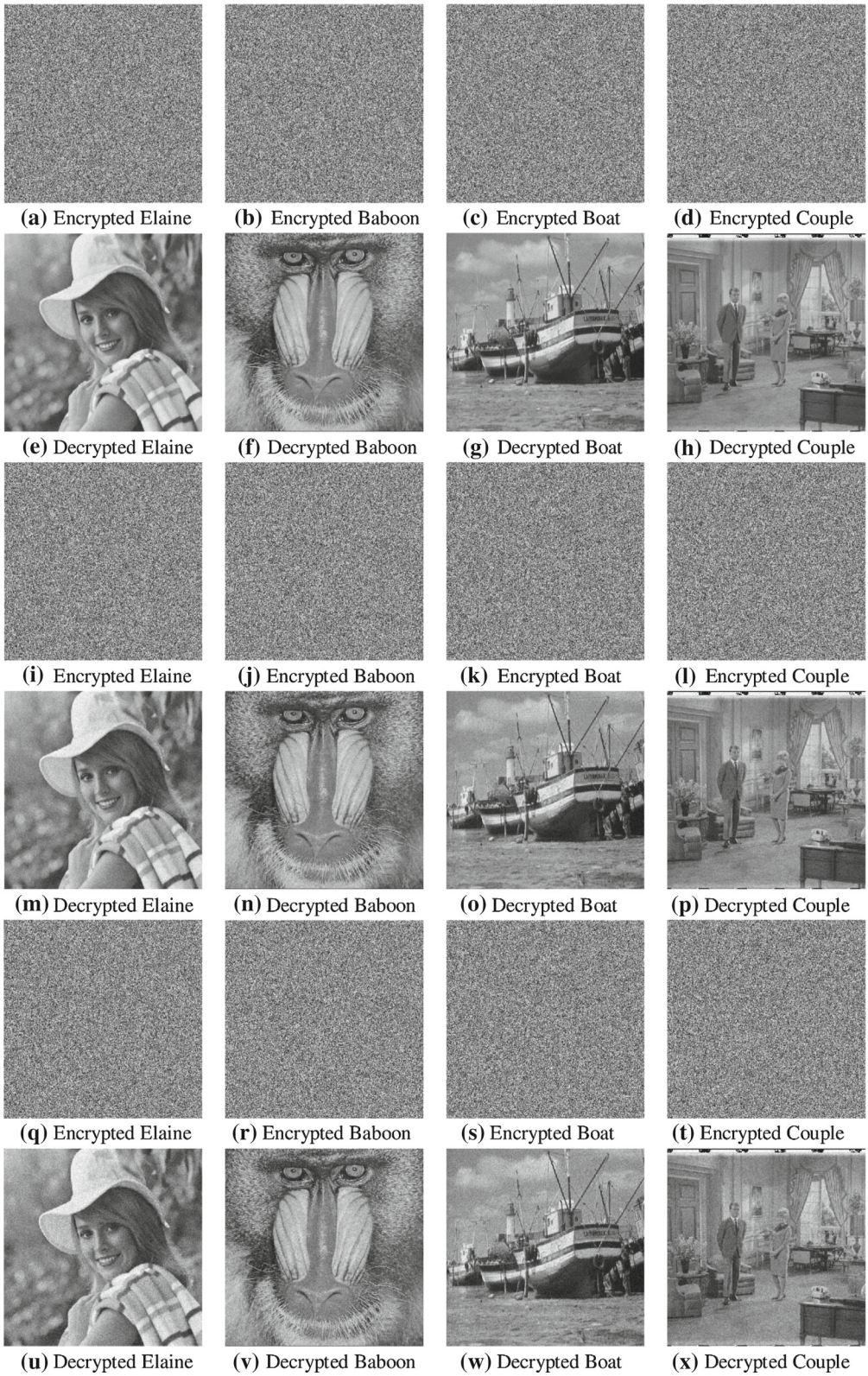
(e) Decrypted Lena (f) Decrypted Baboon (g) Decrypted Pepper (h) Decrypted Sailboat

(i) Encrypted Lena (j) Encrypted Baboon (k) Encrypted Pepper (l) Encrypted Sailboat

(m) Decrypted Lena (n) Decrypted Baboon (o) Decrypted Pepper (p) Decrypted Sailboat

(q) Encrypted Lena (r) Encrypted Baboon (s) Encrypted Pepper (t) Encrypted Sailboat

(u) Decrypted Lena (v) Decrypted Baboon (w) Decrypted Pepper (x) Decrypted Sailboat



(a) Encrypted Elaine (b) Encrypted Baboon (c) Encrypted Boat (d) Encrypted Couple

(e) Decrypted Elaine (f) Decrypted Baboon (g) Decrypted Boat (h) Decrypted Couple

(i) Encrypted Elaine (j) Encrypted Baboon (k) Encrypted Boat (l) Encrypted Couple

(m) Decrypted Elaine (n) Decrypted Baboon (o) Decrypted Boat (p) Decrypted Couple

(q) Encrypted Elaine (r) Encrypted Baboon (s) Encrypted Boat (t) Encrypted Couple

(u) Decrypted Elaine (v) Decrypted Baboon (w) Decrypted Boat (x) Decrypted Couple

◀**Fig. 15** Gaussian noise attack analysis results of Group-2 images: At mean = 0 and variance = 0.0001 **a–d** encrypted images, **e–h** decrypted images; at mean = 0 and variance = 0.0003 **i–l** encrypted images, **m–p** decrypted images; at mean = 0 and variance = 0.0005 **q–t** encrypted images, **u–x** decrypted images

4.11.1 Chosen-plaintext attack

In this type of attack, the attacker has the ability to choose to encrypt some random plaintexts and get their corresponding ciphertexts. In a simple way, with the unknown encryption key K, the attacker has the ciphertext C and tries to obtain the corresponding plaintext P. Even so, with the same unknown encryption key, the attacker has a plaintext P_{az} of all-one (or

Table 25 Gaussian noise attack analysis results of Group-1 images

Between images	Variance	NPCR (%)	UACI (%)
Figures 3(i)–14(e) (First image: "Lena")	0.0001	86.63	4.12
Figures 3(j)–14(f) (Second image: "Baboon")		86.79	4.23
Figures 3(k)–14(g) (Third image: "Pepper")		86.89	4.29
Figures 3(l)–14(h) (Fourth image: "Sailboat")		86.94	4.34
Average		86.8125	4.245
Figures 3(i)–14(m) (First image: "Lena")	0.0003	91.84	5.93
Figures 3(j)–14(n) (Second image: "Baboon")		91.96	6.04
Figures 3(k)–14(o) (Third image: "Pepper")		92.05	6.27
Figures 3(l)–14(p) (Fourth image: "Sailboat")		92.12	6.35
Average		91.9925	6.1475
Figures 3(i)–14(u) (First image: "Lena")	0.0005	93.43	7.12
Figures 3(j)–14(v) (Second image: "Baboon")		93.59	7.36
Figures 3(k)–14(w) (Third image: "Pepper")		93.89	7.41
Figures 3(l)–14(x) (Fourth image: "Sailboat")		94.03	7.48
Average		93.735	7.3425

Table 26 Gaussian noise attack analysis results of Group-2 images

Between images	Noise	NPCR (%)	UACI (%)
Figures 4(i)–15(e) (First image: "Elaine")	0.0001	86.97	4.26
Figures 4(j)–15(f) (Second image: "Baboon")		87.11	4.29
Figures 4(k)–15(g) (Third image: "Boat")		87.24	4.33
Figures 4(l)–15(h) (Fourth image: "Couple")		87.39	4.41
Average		87.1775	4.3225
Figures 4(i)–15(m) (First image: "Elaine")	0.0003	91.96	6.07
Figures 4(j)–15(n) (Second image: "Baboon")		92.01	6.12
Figures 4(k)–15(o) (Third image: "Boat")		92.26	6.29
Figures 4(l)–15(p) (Fourth image: "Couple")		92.37	6.41
Average		92.15	6.2225
Figures 4(i)–15(u) (First image: "Elaine")	0.0005	93.61	7.34
Figures 4(j)–15(v) (Second image: "Baboon")		93.72	7.45
Figures 4(k)–15(w) (Third image: "Boat")		93.96	7.57
Figures 4(l)–15(x) (Fourth image: "Couple")		94.25	7.61
Average		93.885	7.4925

Table 27 Comparison of Gaussian noise attack analysis results

Algorithms	Mean = 0, Variance = 0.0001		Mean = 0, Variance = 0.0003		Mean = 0, Variance = 0.0005	
	NPCR (%)	UACI (%)	NPCR (%)	UACI (%)	NPCR (%)	UACI (%)
Proposed (average of Group-1 images)	86.8125	4.245	91.9925	6.1475	93.735	7.3425
Proposed (average of Group-2 images)	87.1775	4.3225	92.15	6.2225	93.885	7.4925
Patro et al. [50]	88.57	4.69	92.69	6.74	94.60	7.91
Chai et al. [47]	87.7	17.3	93.3	20.3	94.9	21.5
Liu and Wang [61]	99.20	28.44	99.61	28.64	99.61	28.80

all-zero), and its encrypted variant C_{az} acquired. The following is the sub-key extraction by the attacker for pixel encryption [62].

$$K_{az}^{i,j} = C_{az}^{i,j} \oplus P_{az}^{i,j} \tag{23}$$

In Eq. (23), $P_{az}^{i,j}$ is the plaintext with all-zero grayscale pixel values, $C_{az}^{i,j}$ is the corresponding ciphertext obtained with the same unknown key, (i,j) is denoted as the 2D-pixel positions. The operation of Eq. (23) obtains a key stream $K_{az}^{i,j}$.

By using the key stream $K_{az}^{i,j}$, the plaintext $P^{i,j}$ is obtained from the ciphertext $C^{i,j}$ by [62],

$$P^{i,j} = C^{i,j} \oplus K_{az}^{i,j} \tag{24}$$

Figure 18 shows the chosen-plaintext attack on the Group-1 and Group-2 encrypted images using the null-images (all-zero pixel values). By analyzing the chosen-plaintext attack of Group-1 and Group-2 images and their corresponding histograms, it is noticed that the chosen-plaintext fails in this proposed multiple image encryption algorithm. This is because the generated key values of the proposed algorithm are linked to the key values given and to the hash values of the actual images. The proposed algorithm, therefore, has a strong ability to resist the chosen-plaintext attack.

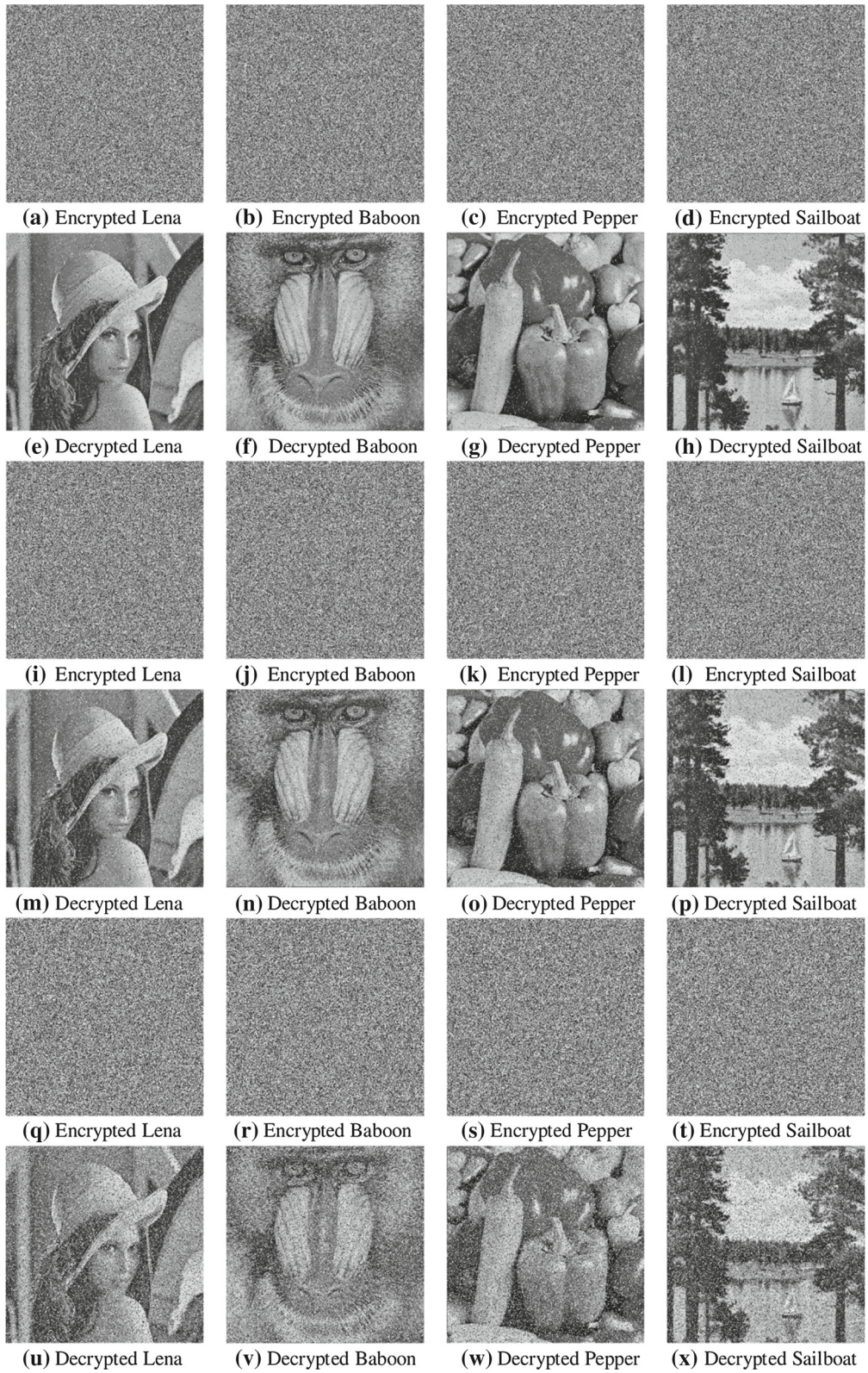
4.11.2 Chosen-ciphertext attack

In this type of attack also, the attacker does not have any key. The attacker has a ciphertext C_{az} of all-one (or all-zero), and its decrypted variant P_{az}. By using Eq. (23), the attacker determines the key stream $K_{az}^{i,j}$. Then using Eq. (24), the plaintext $P^{i,j}$ is recovered from its ciphertext $C^{i,j}$ [62]. Figure 19 shows the chosen-ciphertext attack on the Group-1 and Group-2 images using the null images (all-zero pixel values). By analyzing the chosen-ciphertext attack of Group-1 and Group-2 images and their corresponding histograms, it is noticed that the chosen-ciphertext attack fails in this proposed multiple image encryption algorithm.

4.12 Computational complexity analysis

The computational complexity of the proposed algorithm mainly depends on block permutation operation, image diffusion operation, and PWLCM system-based sequence formation operation. The computational complexity of each of them is described below.

i. *PWLCM system-based chaotic sequence formation operation* Two cross-coupling operations are performed within this algorithm. The first is between PWLCM system-1 and system-2 and the second



(a) Encrypted Lena (b) Encrypted Baboon (c) Encrypted Pepper (d) Encrypted Sailboat

(e) Decrypted Lena (f) Decrypted Baboon (g) Decrypted Pepper (h) Decrypted Sailboat

(i) Encrypted Lena (j) Encrypted Baboon (k) Encrypted Pepper (l) Encrypted Sailboat

(m) Decrypted Lena (n) Decrypted Baboon (o) Decrypted Pepper (p) Decrypted Sailboat

(q) Encrypted Lena (r) Encrypted Baboon (s) Encrypted Pepper (t) Encrypted Sailboat

(u) Decrypted Lena (v) Decrypted Baboon (w) Decrypted Pepper (x) Decrypted Sailboat

◀**Fig. 16** Salt and pepper noise attack analysis results of Group-1 images: At 5% **a–d** encrypted images, **e–h** decrypted images; At 10% **i–l** encrypted images, **m–p** decrypted images; At 25% **q–t** encrypted images, **u–x** decrypted images

cross-coupling is between PWLCM system-3 system-4. The computational complexity for generating the first cross-coupling sequence is $O(2 \times M_g \times N_g)$, and the computational complexity for generating the second cross-coupling sequence is $O(2 \times M_g \times N_g)$. The total computational complexity of the suggested method for the generation of cross-coupling sequences is therefore $O(2M_gN_g + 2M_gN_g) = O(4M_gN_g) \approx O(M_g \times N_g)$.

ii. *Block permutation operation* In the proposed algorithm, the blocks are permuted two times. Each time the number of blocks permuted is $M_g \times N_g$. The computational complexity is therefore $O(M_g \times N_g)$ in each time block permutation operation. The total computational complexity is $O(M_gN_g + M_gN_g) = O(2M_gN_g) \approx O(M_g \times N_g)$. In this algorithm, L–R and U–D flip operations are also performed together with the block permutation operation. In only a few clock cycles, flip processes in existing CPU architectures are completed. Flipping processes do not require constant time; in existing CPU architectures, it is one of the quickest processes.

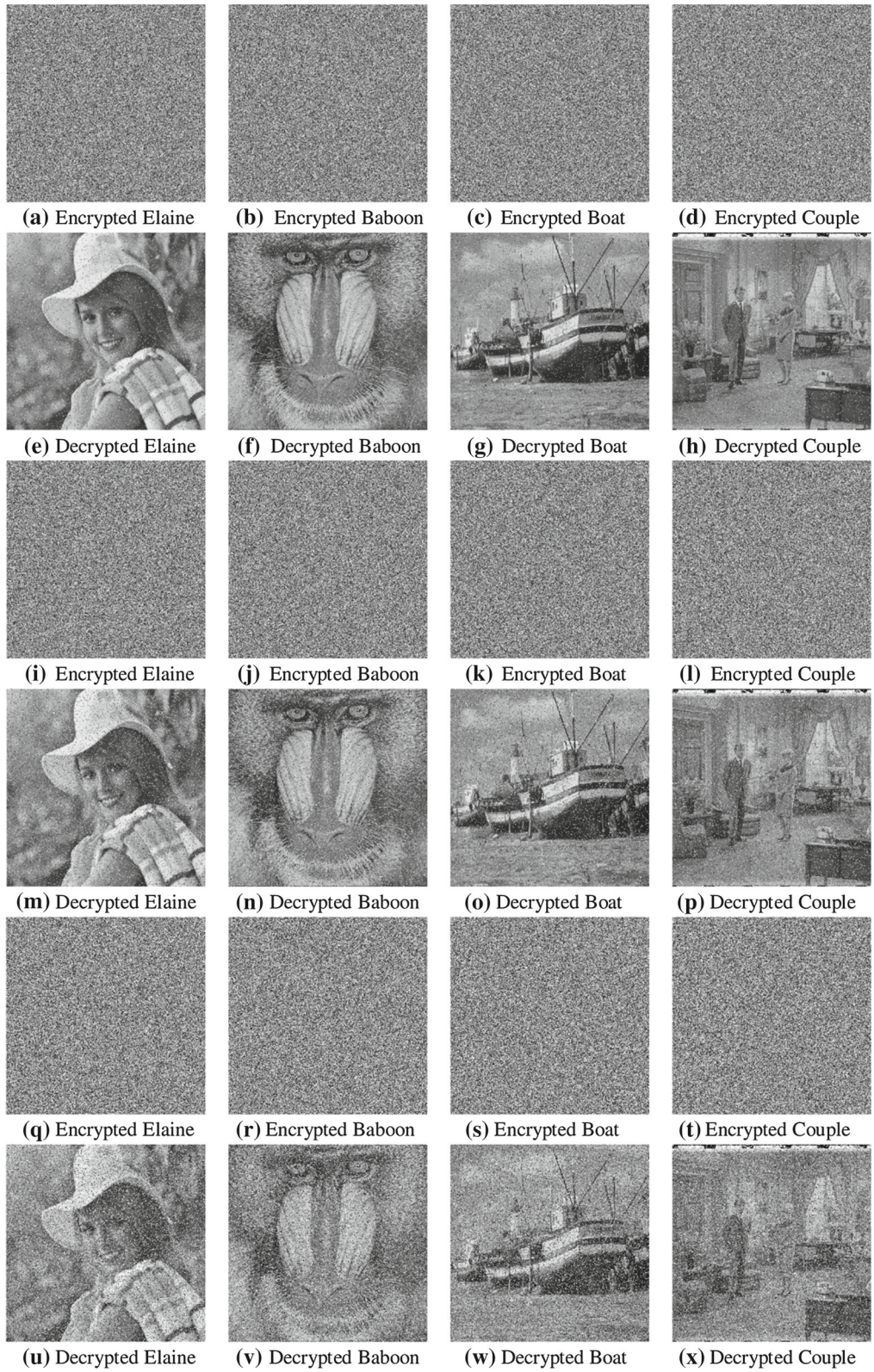
iii. *Image diffusion operation* Pixel-wise bit-XOR-based diffusion operations are executed in the suggested scheme. The computational complexity to perform the process of bit-XOR-based diffusion is $O(n)$, where n represents the counting of bits. In this algorithm, two-time bit-XOR operation is executed between pixels of $M_g \times N_g$, i.e., $M_g \times N_g \times 8$ bits. In each bit-XOR operation, the computational complexity is $O(8 \times M_g \times N_g)$. Hence, the total computational complexity is $O(8M_gN_g + 8M_gN_g) = O(16M_gN_g) \approx O(M_g \times N_g)$.

Therefore, the overall computational complexity of the suggested method is approximately $O(M_g \times N_g)$.

Table 31 provides a comparative analysis of computational complexity between the method of Ref. [25–27, 29, 39] and the suggested method. By observing Table 31, we can found that the proposed method has lesser computational complexity than the third and second algorithms of Zhang and Wang, Tang et al.'s algorithm, and Patro et al.'s algorithm, whereas the proposed algorithm has more computational complexity than the first algorithm of Zhang and Wang. This shows that the suggested method is more efficient than the existing methods of multi-image encryption. However, the computational complexity of the proposed multiple image encryption algorithm is higher than the RC5 and chaotic map-based single-image encryption algorithm developed by Amin and Abd El-Latif et al. [67]. This is because the algorithm in [67] operates on a fixed block size and takes around the same time regardless of input, so the computational complexity is $O(1)$. But we typically get an $O(m)$ complexity for the encryption of longer messages using the mode of operation, where m is the number of data blocks to be encrypted.

4.13 Comparison of the permutation operations

The comparison of the permutation operation between the proposed algorithm and the algorithm in Ref. [68] is provided in Table 32. Similarly, the comparison of the permutation operation between the proposed algorithm and the algorithm in Ref. [69] is provided in Table 33.



(a) Encrypted Elaine (b) Encrypted Baboon (c) Encrypted Boat (d) Encrypted Couple

(e) Decrypted Elaine (f) Decrypted Baboon (g) Decrypted Boat (h) Decrypted Couple

(i) Encrypted Elaine (j) Encrypted Baboon (k) Encrypted Boat (l) Encrypted Couple

(m) Decrypted Elaine (n) Decrypted Baboon (o) Decrypted Boat (p) Decrypted Couple

(q) Encrypted Elaine (r) Encrypted Baboon (s) Encrypted Boat (t) Encrypted Couple

(u) Decrypted Elaine (v) Decrypted Baboon (w) Decrypted Boat (x) Decrypted Couple

◀**Fig. 17** Salt and pepper noise attack analysis results of Group-2 images: At 5% **a–d** encrypted images, **e–h** decrypted images; at 10% **i–l** encrypted images, **m–p** decrypted images; at 25% **q–t** encrypted images, **u–x** decrypted images

5 Conclusion

This paper proposes an efficient two-layer security-based multi-image encryption scheme. In this scheme,

Table 28 Salt and pepper noise attack analysis results of Group-1 images

Between images	Noise (%)	NPCR (%)	UACI (%)
Figures 3(i)–16(e) (First image: "Lena")	5	23.96	3.16
Figures 3(j)–16(f) (Second image: "Baboon")		24.07	3.21
Figures 3(k)–16(g) (Third image: "Pepper")		24.12	3.24
Figures 3(l)–16(h) (Fourth image: "Sailboat")		24.23	3.36
Average		24.095	3.2425
Figures 3(i)–16(m) (First image: "Lena")	10	42.87	4.02
Figures 3(j)–16(n) (Second image: "Baboon")		43.14	4.19
Figures 3(k)–16(o) (Third image: "Pepper")		43.21	4.23
Figures 3(l)–16(p) (Fourth image: "Sailboat")		43.26	4.46
Average		43.12	4.225
Figures 3(i)–16(u) (First image: "Lena")	25	55.98	11.23
Figures 3(j)–16(v) (Second image: "Baboon")		56.02	11.27
Figures 3(k)–16(w) (Third image: "Pepper")		56.29	11.40
Figures 3(l)–16(x) (Fourth image: "Sailboat")		56.32	11.46
Average		56.1525	11.34

Table 29 Salt and pepper noise attack analysis results of Group-2 images

Between images	Noise (%)	NPCR (%)	UACI (%)
Figures 4(i)–17(e) (First image: "Elaine")	5	24.01	3.21
Figures 4(j)–17(f) (Second image: "Baboon")		24.11	3.27
Figures 4(k)–17(g) (Third image: "Boat")		24.25	3.31
Figures 4(l)–17(h) (Fourth image: "Couple")		24.34	3.40
Average		24.1775	3.2975
Figures 4(i)–17(m) (First image: "Elaine")	10	42.88	4.11
Figures 4(j)–17(n) (Second image: "Baboon")		43.11	4.23
Figures 4(k)–17(o) (Third image: "Boat")		43.23	4.33
Figures 4(l)–17(p) (Fourth image: "Couple")		43.31	4.41
Average		43.1325	4.27
Figures 4(i)–17(u) (First image: "Elaine")	25	56.08	11.51
Figures 4(j)–17(v) (Second image: "Baboon")		56.17	11.60
Figures 4(k)–17(w) (Third image: "Boat")		56.33	11.68
Figures 4(l)–17(x) (Fourth image: "Couple")		56.42	11.73
Average		56.25	11.63

Table 30 Comparison of salt and pepper noise attack analysis results

Algorithms	Noise = 5%		Noise = 10%		Noise = 25%	
	NPCR (%)	UACI (%)	NPCR (%)	UACI (%)	NPCR (%)	UACI (%)
Proposed (average of Group-1 images)	24.095	3.2425	43.12	4.225	56.1525	11.34
Proposed (average of Group-2 images)	24.1775	3.2975	43.1325	4.27	56.25	11.63
Patro et al. [50]	24.50	3.39	43.18	4.47	56.63	11.96

Table 31 Comparison of computational complexity

Algorithms	Computational complexity
Proposed	$\left[\frac{(C_g+S_g)M_gN_g}{L+L}+\frac{(C_g+X_g)M_gN_g}{M_g\times N_g}+C_g\right]L$
Zhang and Wang (third scheme) in Ref. [29]	$\left[L(4C_g+2D_g+S_g)+LX_g+C_g+D_g\right]M_gN_g$
Zhang and Wang (second scheme) in Ref. [29]	$\left[\frac{(C_g+S_g)M_gN_g}{64}+\frac{(C_g+X_g)M_gN_g}{64}+C_g\right]L$
Zhang and Wang (first scheme) in Ref. [29]	$\left[\frac{(C_g+S_g)M_gN_g}{64}+C_g\right]L$
Tang et al. in Ref. [29]	$\left[\frac{S_g}{2}+L(2B_g+X_g)+C_g\right]M_gN_g$
Patro et al. [39]	$\left[\frac{(C_g+S_g)M_gN_g}{M_g+N_g}+\frac{(C_g+X_g)M_gN_g}{M_g+N_g}+C_g\right]L$

Parameters: Size of images = $M_g \times N_g$, Number of images = L

Computational complexity for, pixel encoding/decoding operation = D_g, bit block/pixel block/pixel/DNA code scrambling operation = S_g, pixel conversion operation (binary-to-decimal) = B_g, pixel XOR operation = X_g, one-time chaotic iteration = C_g

two distinct layers of permutation and flip operation and then two distinct layers of diffusion operation are conducted. To achieve permutation and diffusion, a cross-coupling of PWLCM systems is used. Two cross-couplings are carried out in two different layers. This renders the algorithm more efficient. In cross-coupling, the need for a single 1D chaotic map-PWLCM system makes the algorithm efficient, both hardware and software. In additions to that, the block-based permutation, L–R, and U–D flip operations minimize the algorithm's computational complexity. Moreover, the hash generated keys of the algorithm resist both the chosen-plaintext attack and the known-plaintext attack. Results of the simulations show the suggested method is more efficient in encryption. The security review shows that all the security attacks that are widely used are strongly resisted by the suggested scheme. The comparative analysis reveals that the

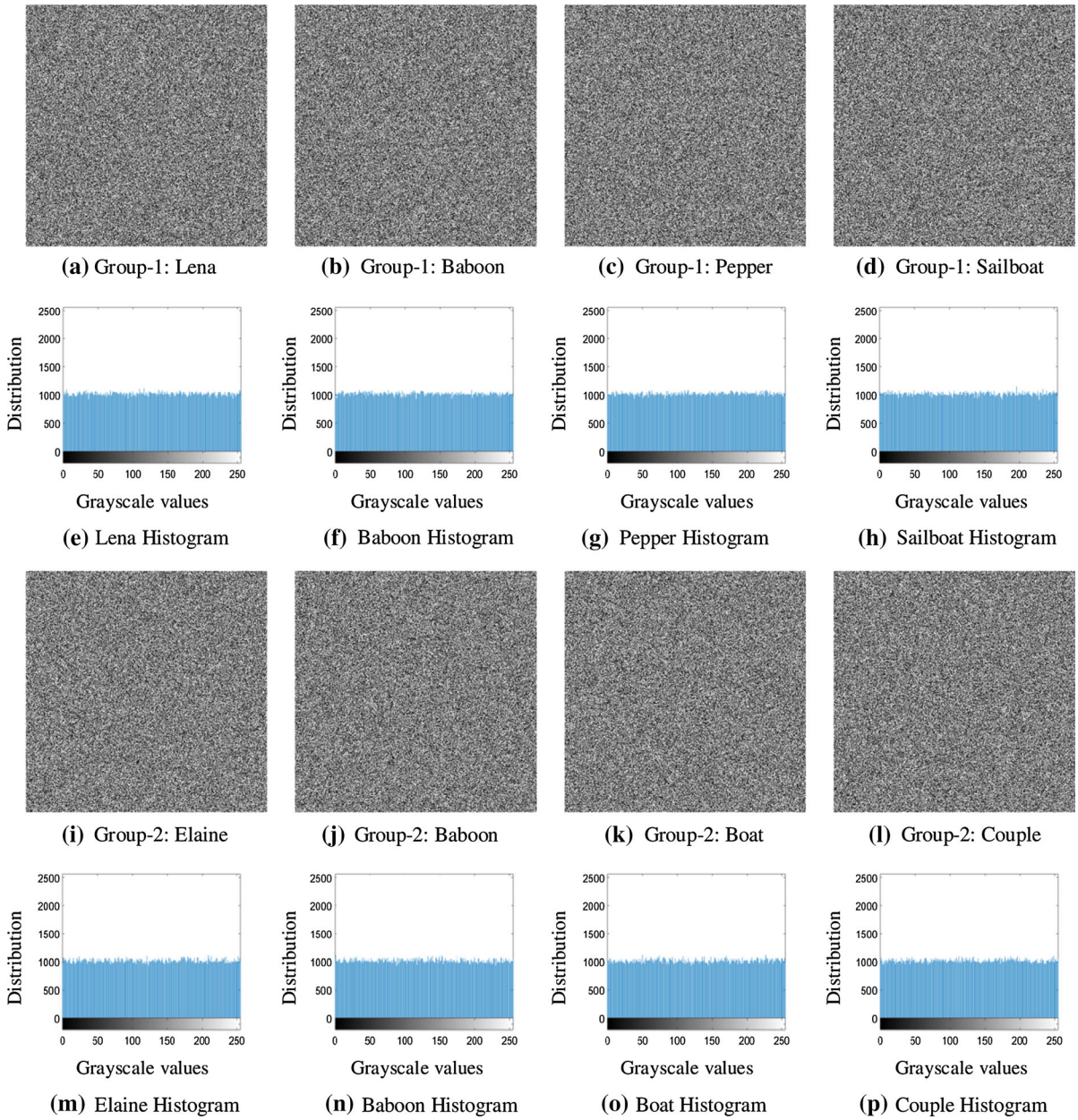


Fig. 18 Cryptanalysis **a**–**d** chosen-plaintext attack of Group-1 images, **e**–**h** corresponding Group-1 image histograms; **i**–**l** chosen-plaintext attack of Group-2 images, **m**–**p** corresponding Group-2 image histograms

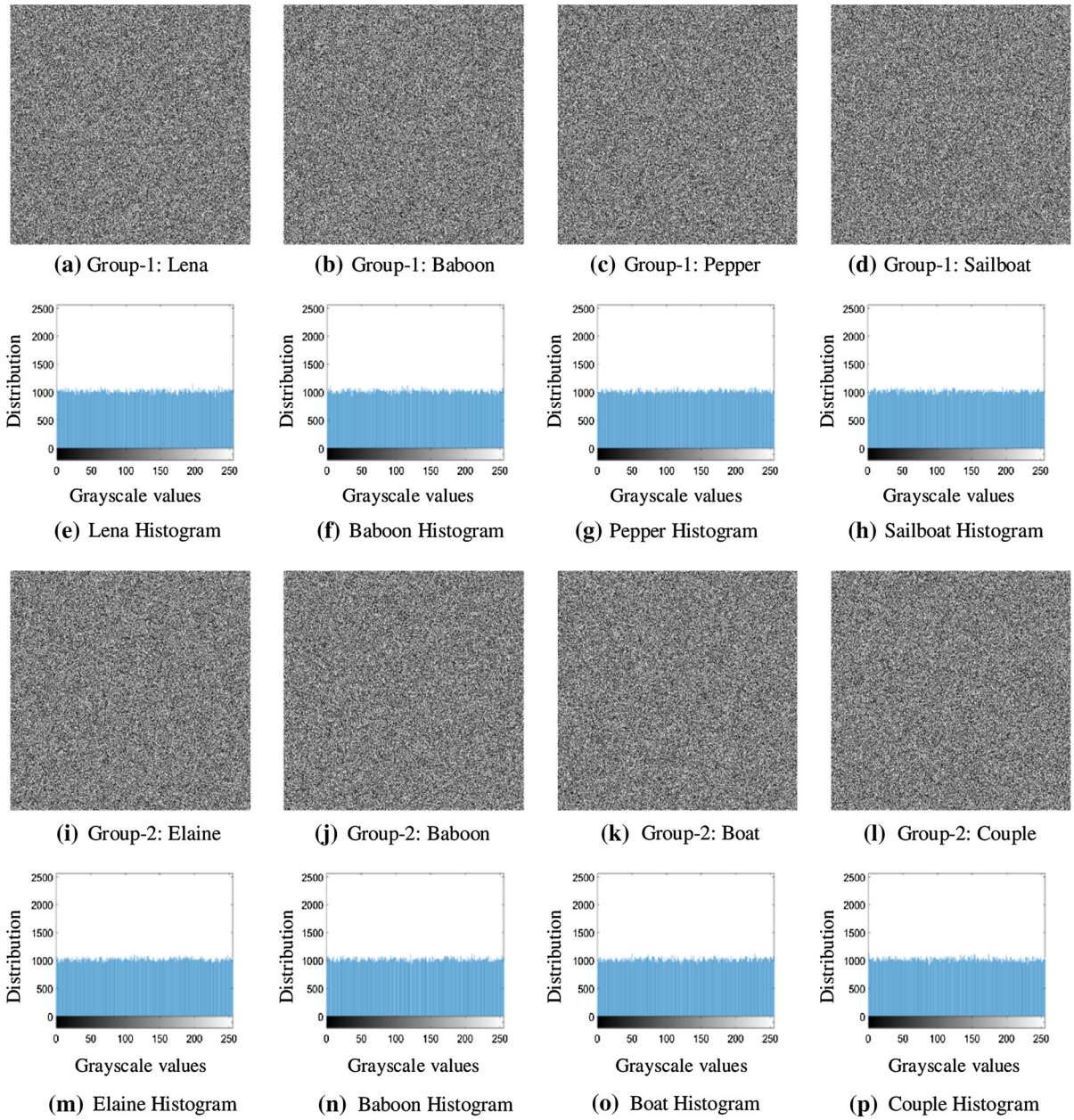


Fig. 19 Cryptanalysis **a**–**d** chosen-ciphertext attack of Group-1 images, **e**–**h** corresponding Group-1 image histograms; **i**–**l** chosen-ciphertext attack of Group-2 images, **m**–**p** corresponding Group-2 image histograms

Table 32 Comparison of permutation operation between the proposed algorithm and Ref [68] algorithm

S. No.	Parameters	Proposed algorithm	Ref. [68] algorithm
1.	Permutation operation	Performs both block- and pixel-level permutation operation. Increases the security of the algorithm	Performs both block- and pixel-level permutation operation. Increases the security of the algorithm
2.	Chaotic system used	Uses cross-coupling of chaotic systems. Cryptanalysis is harder	Uses a single chaotic system. Cryptanalysis is not so harder
3.	Block-level permutation operation	Block-level permutation is performed using the cross-coupled PWLCM systems	Block-level permutation is performed using the generalized cat map
4.	Pixel-level permutation operation	Pixel-level permutation is performed in the form of flip operation (L–R and U–D). In only a few clock cycles, flip processes in existing CPU architectures are completed. Flipping processes do not require constant time; in existing CPU architectures, it is one of the quickest processes	Pixel-level permutation is performed using the chaotic logistic map. The computational complexity of the pixel permutation is $O(N \times m \times m)$, where $m \times m$ is the size of the each block and N is the number of blocks
5	No. of times of permutation operation	Both block- and pixel-level permutation operations are performed two times. More hardening of the cryptanalysis	Both block- and pixel-level permutation operations are performed only one time. Cryptanalysis is not so harder

Table 33 Comparison of permutation operation between the proposed algorithm and Ref. [69] algorithm

S. No.	Parameters	Proposed algorithm	Ref. [69] algorithm
1.	Transform operations	No transform operations are performed in the permutation of pixels. Decreases the computational complexity of the algorithm	Transform operations are performed in the permutation of pixels. Increases the computational complexity of the algorithm. This is because in transform operations, there is a requirement of spatial domain into transform domain and vice versa
2.	Permutation operation	Performs both block- and pixel-level permutation operation. Increases the security of the algorithm	Performs only pixel-level permutation operation
3.	Chaotic system used	Uses cross-coupling of PWLCM systems. Cryptanalysis is harder	Uses a single chaotic system such as 2D chaotic standard map. Cryptanalysis is not so harder
4.	Block-level permutation operation	Block-level permutation is performed using the cross-coupled PWLCM systems	No such block-level permutation is performed
5.	Pixel-level permutation operation	Pixel-level permutation is performed in the form of flip operation (L–R and U–D). In only a few clock cycles, flip processes in existing CPU architectures are completed. Flipping processes do not require constant time; in existing CPU architectures, it is one of the quickest processes	Pixel-level permutation is performed using the 2D chaotic standard map. The computational complexity of the pixel-level permutation is $O(M \times N)$, where $M \times N$ is the size of the image
6.	No. of times of permutation operation	Both block- and pixel-level permutation operations are performed two times. More hardening of the cryptanalysis	Pixel-level permutation operation is performed only once. Cryptanalysis is not so harder

suggested method for multi-image encryption is stronger and more effective than the current methods for multi-image encryption.

Funding There is no funding received by any funding agency.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Coppersmith, D.: The data encryption standard (DES) and its strength against attacks. IBM J. Res. Dev. **38**(3), 243–250 (1994)
2. Advanced Encryption Standard (AES): N.F. Pub, 197, Federal information processing standards publication. **197**(441), 0311 (2001)
3. Gao, H., Zhang, Y., Liang, S., Li, D.: A new chaotic algorithm for image encryption. Chaos, Solitons Fractals **29**(2), 393–399 (2006)
4. Patro, K.A.K., Acharya, B., Nath, V.: A secure multi-stage one-round bit-plane permutation operation based chaotic image encryption. Microsyst. Technol. **25**(6), 2331–2338 (2019)
5. Guesmi, R., Farah, M.A.B., Kachouri, A., Samet, M.: A novel chaos-based image encryption using DNA sequence operation and Secure Hash Algorithm SHA-2. Nonlinear Dyn. **83**(3), 1123–1136 (2016)
6. Guesmi, R., Farah, M.A.B., Kachouri, A., Samet, M.: Hash key-based image encryption using crossover operator and chaos. Multimed. Tools Appl. **75**(8), 4753–4769 (2016)
7. Patro, K.A.K., Acharya, B.: A simple, secure, and time-efficient bit-plane operated bit-level image encryption scheme using 1-D chaotic maps. In: Chattopadhyay, J., Singh, R., Bhattacherjee, V. (eds.) Innovations in Soft Computing and Information Technology, pp. 261–278. Springer (2019)
8. Kulsoom, A., Xiao, D., Abbas, S.A.: An efficient and noise resistive selective image encryption scheme for gray images based on chaotic maps and DNA complementary rules. Multimed. Tools Appl. **75**(1), 1–23 (2016)
9. Patro, K.A.K., Babu, M.P.J., Kumar, K.P., Acharya, B.: Dual-layer DNA-encoding–decoding operation based image encryption using one-dimensional chaotic map. In: Kolhe, M.L., Tiwari, S., Trivedi, M.C., Mishra, K.K. (eds.) Advances in Data and Information Sciences, pp. 67–80. Springer (2020)
10. Wang, T., Wang, M.H.: Hyperchaotic image encryption algorithm based on bit-level permutation and DNA encoding. Opt. Laser Technol. **132**, 106355 (2020)
11. Patro, K.A.K., Acharya, B.: A secure block operation based bit-plane image encryption using chaotic maps. In: IEEE First International Conference on Power, Control and Computing Technologies (ICPC2T), pp. 411–416 (2020)
12. Abdelfatah, R.I.: A new fast double-chaotic based Image encryption scheme. Multimed. Tools Appl. **79**(1), 1241–1259 (2020)
13. Liu, W., Sun, K., Zhu, C.: A fast image encryption algorithm based on chaotic map. Opt. Lasers Eng. **84**, 26–36 (2016)
14. Wang, X., Wang, S., Zhang, Y., Guo, K.: A novel image encryption algorithm based on chaotic shuffling method. Inf. Secur. J. A Glob. Perspect. **26**(1), 7–16 (2017)
15. Behnia, S., Akhshani, A., Mahmodi, H., Akhavan, A.: A novel algorithm for image encryption based on mixture of chaotic maps. Chaos, Solitons Fractals **35**(2), 408–419 (2008)
16. Sneha, P.S., Sankar, S., Kumar, A.S.: A chaotic colour image encryption scheme combining Walsh–Hadamard transform and Arnold-Tent maps. J. Ambient. Intell. Humaniz. Comput. **11**(3), 1289–1308 (2020)
17. El-Latif, A.A.A., Li, L., Zhang, T., Wang, N., Song, X., Niu, X.: Digital image encryption scheme based on multiple chaotic systems. Sens. Imaging Int. J. **13**(2), 67–88 (2012)
18. Liu, H., Xu, Y., Ma, C.: Chaos based image hybrid encryption algorithm using key stretching and hash feedback. Optik **216**, 164925 (2020)
19. Lu, Q., Zhu, C., Deng, X.: An efficient image encryption scheme based on the LSS chaotic map and single S-box. IEEE Access **8**, 25664–25678 (2020)
20. Patro, K.A.K., Acharya, B., Nath, V.: Various dimensional colour image encryption based on non-overlapping block-level diffusion operation. Microsyst. Technol. **26**, 1437–1448 (2020)
21. Gan, Z., Chai, X., Zhang, M., Lu, Y.: A double color image encryption scheme based on three-dimensional brownian motion. Multimed. Tools Appl. **77**, 27919–27953 (2018)
22. Singh, N., Sinha, A.: Chaos based multiple image encryption using multiple canonical transforms. Opt. Laser Technol. **42**(5), 724–731 (2010)
23. Kong, D., Shen, X., Xu, Q., Xin, W., Guo, H.: Multiple-image encryption scheme based on cascaded fractional fourier transform. Appl. Opt. **52**(12), 2619–2625 (2013)
24. Kong, D., Shen, X.: Multiple-image encryption based on optical wavelet transform and multichannel fractional fourier transform. Opt. Laser Technol. **57**, 343–349 (2014)
25. Tang, Z., Song, J., Zhang, X., Sun, R.: Multiple-image encryption with bit-plane decomposition and chaotic maps. Opt. Lasers Eng. **80**, 1–11 (2016)
26. Zhang, X., Wang, X.: Multiple-image encryption algorithm based on mixed image element and permutation. Opt. Lasers Eng. **92**, 6–16 (2017)
27. Zhang, X., Wang, X.: Multiple-image encryption algorithm based on mixed image element and chaos. Comput. Electr. Eng. **62**, 401–413 (2017)
28. Li, C.-L., Li, H.-M., Li, F.-D., Wei, D.-Q., Yang, X.-B., Zhang, J.: Multiple-image encryption by using robust chaotic map in wavelet transform domain. Optik **171**, 277–286 (2018)
29. Zhang, X., Wang, X.: Multiple-image encryption algorithm based on dna encoding and chaotic system. Multimed. Tools Appl. **78**(6), 7841–7869 (2019)
30. Enayatifar, R., Guimarães, F.G., Siarry, P.: Index-based permutation-diffusion in multiple-image encryption using DNA sequence. Opt. Lasers Eng. **115**, 131–140 (2019)

31. Shujun, L., Xuanqin, M., Yuanlong, C.: Pseudo-random bit generator based on couple chaotic systems and its applications in streamcipher cryptography. In: International Conference on Cryptology in India, Springer, pp. 316–329 (2001)
32. Short, K.M.: Signal extraction from chaotic communications. Int. J. Bifurc. Chaos **7**(07), 1579–1597 (1997)
33. Yang, T., Yang, L.-B., Yang, C.-M.: Cryptanalyzing chaotic secure communications using return maps. Phys. Lett. A **245**(6), 495–510 (1998)
34. Ogorzatek, M. J., Dedieu, H.: Some tools for attacking secure communication systems employing chaotic carriers. In: ISCAS'98. Proceedings of the 1998 IEEE International Symposium on Circuits and Systems (Cat. No. 98CH36187), Vol. 4, IEEE, pp. 522–525 (1998)
35. Zhou, C.-S., Chen, T.-L.: Extracting information masked by chaos and contaminated with noise: some considerations on the security of communication approaches using chaos. Phys. Lett. A **234**(6), 429–435 (1997)
36. Beth, T., Lazic, D. E., Mathias, A.: Cryptanalysis of cryptosystems based on remote chaos replication. In: Annual International Cryptology Conference, pp. 318–331. Springer (1994)
37. Heidari-Bateni, G., McGillem, C.D.: A chaotic direct-sequence spread-spectrum communication system. IEEE Trans. Commun. **42**(234), 1524–1527 (1994)
38. Protopopescu, V.A., Santoro, R.T., Tolliver, J. S.: Fast and secure encryption-decryption method based on chaotic dynamics, US Patent 5479513 (1995)
39. Patro, K.A.K., Soni, A., Netam, P.K., Acharya, B.: Multiple grayscale image encryption using cross-coupled chaotic maps. J. Inf. Secur. Appl. **52**, 102470 (2020)
40. Xiang, T., Liao, X., Wong, K.W.: An improved particle swarm optimization algorithm combined with piecewise linear chaotic map. Appl. Math. Comput. **190**(2), 1637–1645 (2007)
41. Usc-sipi image database for research in image processing, image analysis, and machine vision, http://sipi.usc.edu/database/ (Accessed 19 Sep 2017)
42. Li, Y., Wang, C., Chen, H.: A hyper-chaos-based image encryption algorithm using pixel-level permutation and bit-level permutation. Opt. Lasers Eng. **90**, 238–246 (2017)
43. Sravanthi, D., Patro, K.A.K., Acharya, B., Majumder, S.: A secure chaotic image encryption based on bit-plane operation. In: Soft computing in Data Analytics, pp. 717–726. Springer (2019)
44. I. C. S. S. C. W. Group of the Microprocessor Standards Subcommittee, IEEE standard for binary floating-point arithmetic, Institute of Electrical and Electronic Engineers (1985)
45. Chen, J.X., Zhu, Z.L., Fu, C., Yu, H., Zhang, L.B.: A fast chaos-based image encryption scheme with a dynamic state variables selection mechanism. Commun. Nonlinear Sci. Numer. Simul. **20**(3), 846–860 (2015)
46. Patro, K.A.K., Acharya, B.: A novel multi-dimensional multiple image encryption technique. Multimed. Tools Appl. **79**, 12959–12994 (2020)
47. Chai, X., Chen, Y., Broyde, L.: A novel chaos-based image encryption algorithm using DNA sequence operations. Opt. Lasers Eng. **88**, 197–213 (2017)
48. Sravanthi, D., Patro, K.A.K., Acharya, B., Babu, M.P.J.: Simple permutation and diffusion operation based image encryption using various one-dimensional chaotic maps: a comparative analysis on security. In: Advances in Data and Information Sciences, pp. 81–96. Springer (2020)
49. Liao, X., Hahsmi, M.A., Haider, R.: An efficient mixed inter-intra pixels substitution at 2bits-level for image encryption technique using DNA and chaos. Optik-Int. J. Light Electron Opt. **153**, 117–134 (2018)
50. Patro, K.A.K., Acharya, B., Nath, V.: Secure, lossless, and noise-resistive image encryption using chaos, hyper-chaos, and DNA sequence operation. IETE Tech. Rev. **37**(3), 223–245 (2020)
51. Kwok, H.S., Tang, W.K.: A fast image encryption system based on chaotic maps with finite precision representation. Chaos, Solitons Fractals **32**(4), 1518–1529 (2007)
52. Patro, K.A.K., Acharya, B.: Novel data encryption scheme using DNA computing. In: Advances of DNA Computing in Cryptography, pp. 69–110. Chapman and Hall/CRC (2018)
53. Norouzi, B., Mirzakuchaki, S., Seyedzadeh, S.M., Mosavi, M.R.: A simple, sensitive and secure image encryption algorithm based on hyper-chaotic system with only one round diffusion process. Multimed. Tools Appl. **71**(3), 1469–1497 (2014)
54. Patro, K.A.K., Acharya, B., Nath, V.: Secure multilevel permutation-diffusion based image encryption using chaotic and hyper-chaotic maps. Microsyst. Technol. **25**(12), 4593–4607 (2019)
55. Samiullah, M., Aslam, W., Nazir, H., Lali, M.I., Shahzad, B., Mufti, M.R., Afzal, H.: An image encryption scheme based on DNA computing and multiple chaotic systems. IEEE Access **8**, 25650–25663 (2020)
56. Tsafack, N., Kengne, J., Abd-El-Atty, B., Iliyasu, A.M., Hirota, K., A.A. Abd EL-Latif, : Design and implementation of a simple dynamical 4-D chaotic circuit with applications in image encryption. Inf. Sci. **515**, 191–217 (2020)
57. Abd El-Latif, A.A., Abd-El-Atty, B., Mazurczyk, W., Fung, C., Venegas-Andraca, S.E.: Secure data encryption based on quantum walks for 5G Internet of Things scenario. IEEE Trans. Netw. Serv. Manag. **17**(1), 118–131 (2020)
58. Sambas, A., Vaidyanathan, S., Tlelo-Cuautle, E., Abd-El-Atty, B., Abd El-Latif, A.A., Guillén-Fernández, O., Hidayat, Y., Gundara, G.: A 3-D multi-stable system with a peanut-shaped equilibrium curve: circuit design FPGA realization, and an application to image encryption. IEEE Access **8**, 137116–137132 (2020)
59. Malik, D.S., Shah, T.: Color multiple image encryption scheme based on 3D-chaotic maps. Math. Comput. Simul. **178**, 646 (2020)
60. Zhu, C.: A novel image encryption scheme based on improved hyperchaotic sequences. Opt. Commun. **285**(1), 29–37 (2012)
61. Liu, H., Wang, X.: Image encryption using DNA complementary rule and chaotic maps. Appl. Soft Comput. **12**(5), 1457–1466 (2012)
62. Nkandeu, Y.P.K., Tiedeu, A.: An image encryption algorithm based on substitution technique and chaos mixing. Multimed. Tools Appl. **78**(8), 10013–10034 (2019)
63. Benrhouma, O., Hermassi, H., Abd El-Latif, A.A., Belghith, S.: Cryptanalysis of a video encryption method based on

mixing and permutation operations in the DCT domain. Signal Image Video Process. **9**(6), 1281–1286 (2015)
64. R. Bechikh, H. Hermassi, A.A. Abd El-Latif, R. Rhouma, S. Belghith, Breaking an image encryption scheme based on a spatiotemporal chaotic system, Signal Processing: Image Communication 39 (2015) 151–158.
65. Zhang, X., Nie, W., Ma, Y., Tian, Q.: Cryptanalysis and improvement of an image encryption algorithm based on hyper-chaotic system and dynamic S-box. Multimed. Tools Appl. **76**(14), 15641–15659 (2017)
66. Fan, H., Li, M., Liu, D., An, K.: Cryptanalysis of a plaintext-related chaotic RGB image encryption scheme using total plain image characteristics. Multimedia Tools and Applications **77**(15), 20103–20127 (2018)
67. Amin, M., Abd El-Latif, A.A.: Efficient modified RC5 based on chaos adapted to image encryption. J. Electron. Imaging **19**(1), 013012 (2010)
68. Belazi, A., Abd El-Latif, A.A., Rhouma, R., Belghith, S.: Selective image encryption scheme based on DWT, AES S-box and chaotic permutation, in: International wireless communications and mobile computing conference (IWCMC), pp. 606–610. IEEE (2015)
69. Abd El-Latif, A.A., Niu, X., Amin, M.: A new image cipher in time and frequency domains. Opt. Commun. **285**(21–22), 4241–4251 (2012)