Journal of Engineering Science and Technology
Vol. 17, No. 3 (2022) 2135 - 2156

COLOUR IMAGE PRIVACY BASED ON CASCADED DESIGN OF SYMMETRIC BLOCK CIPHER

SEERWAN W. JIRJEES*, NOOR A. YOUSIF, ASHWAQ T. HASHIM

Control and Systems Engineering Department, University of Technology-Iraq
*Corresponding Author: seerwan.w.jirjees@uotechnology.edu.iq

Abstract

Image privacy became an important issue due to the image usage increasing in most communications. It ensures the security of information embedded in these images, such as military and medical images. An image has some features like the high correlation among their pixels and high redundancy; thus, most traditional block ciphers are not suitable for image encryption. This paper introduces some new criteria that can be used for enhancing the design of standard block ciphers like the RC6 algorithm. It is designed to meet the requirements of increasing security and better performance for image encryption. The RC6-like block cipher with the 3D permutation model is proposed. It is 2048 bits, and the key is 256 bits. The proposed permutation is a bitwise permutation based on a hyperchaotic system besides two Chebyshev maps to eliminate the correlation of the image pixels. A round of RC6 block cipher is used as F-function in the proposed encryption algorithm. It used a type three Feistel network in cascaded design instead of rounds. It uses a like-chaining process that causes the decryption of a ciphertext block to depend on the preceding ciphertext blocks. The number of rounds is 30 rounds. It can be established through theoretical analysis and experimental results the suggested scheme has excellent opportunities for application in secure image communication.

Keywords: Chaotic system, Colour image encryption, Cryptography, RC6.

1. Introduction

Multimedia information has been increasingly used and shared via the Internet, especially in recent years, including short videos, photos, and voice data, which have various distinctive features to text information easily understood and clearly represented [1]. There are several data protection approaches, including a wide variety of algorithms used for data encryption. However, traditional encryption algorithms proposed for text information such as DES, IDES, And RSA are not suitable for image encryption due to the large data size, high redundancy, and stronger pixel correlation in images [2].

Matthews [3] established the first chaotic stream encryption algorithm. Next, chaotic maps are used to encrypt image, which gets an advantage from their excellent properties, like strong ergodicity besides sensitivity to control parameters and initial conditions [4]; these features correspond to the effected of diffusion and confusion in the cryptography sense. Therefore, the right choice for image encryption is a chaotic system [5]. Unlike traditional cryptographic techniques, chaos-based image encryption schemes have performed well in trade-offs between security and efficiency [6]. Since the early image encryption approach based on chaos was suggested by Fridrich [7]. This structure had later worldwide attention, and at present, most chaos-based image encryption approaches rely on this structure and have attained a suitable cipher effect [8].

The rest of the paper is organized as follows. Section 2 contains reviews on related work. The Proposed algorithm in detail is presented in section 3. Section 4 presents experimental results and performance analysis. Finally, the conclusions are made in Section 5

2. Related Works

In the current literature, many image encryptions on the basis of chaotic map approaches are found. Younesa and Jantan [9] suggested a conversion algorithm called the Blowfish based on a combination of image transformation and a known ciphering and deciphering algorithm. The original image was divided into blocks using the conversion algorithm which were rearranged into a converted image , and then the converted image was encoded using the blowfish algorithm. Zhang and Liu [10] suggested an approach where the P-box is selected as the same size as a plain image, which scrambled pixel positions. The keystream produced through skew tent chaotic map is correlated to the plain image. Liu and Wang [11] suggested a high-dimensional chaotic map for bit-level permutation for colour image encryption.

Boriga et al. [12] presented a chaos-based cipher approach based on three 2D chaotic maps. The proposed approach holds two classical stages: a confusion stage where a random permutation generated by an efficient algorithm is employed to shuffle pixels and a diffusion stage where the values of pixels are changed by a new XOR scheme. Wang et al. [13] proposed a colour image encryption algorithm based on chaos. Because chaotic maps with low dimensional have the gains of simplicity and adequacy, Wang et al. [14] realised a cryptographic scheme for permutated colour images by zigzag track permutation with a chaotic spatiotemporal scheme.

Murillo-Escobar et al. [15] adapted an encryption method for RGB colour image relied on wholly image features. The logistic map is provided. Zheng and

Jin [16] introduced an algorithm of image encryption that relied on Henon map and compound spatiotemporal chaotic with dominance key sensitivity, plaintext sensitivity, and execution efficiency. A simple pixel-based scrambling for image encryption algorithm is developed by Ye [17] to detect alteration of a grey pixel value and its position simultaneously.

Fu et al. [18] utilised an algorithm of bit-level permutation in two phases to shuffle the original image, which is gained the effected of diffusion in the permutation step. But Li et al. [19] noted that any cryptographic systems that use only the permutation operation could efficiently broken with O([$\log_2$MN][$\log_2$MN]) plaintexts and O([$\log_2$MN][$\log_2$MN]×MN2) computational time. Consequently, diffusion is, therefore, necessary to the security of image encryption. The consideration of conventional systems is not the right choice of transmitting a message efficiently; thus, employing chaotic maps becomes necessary in designing a secure cryptosystem.

Lately, high dimensional chaotic maps, besides hybrid chaos systems, are used to algorithms key space expansion and resisting brute force attack. Li et al. [20] suggested an image encryption algorithm based on a hyperchaos system. The algorithm adopts a 5-D hyper scattered system multi-wing, and the key stream generated by the hyper-chaotic system is related to the raw image. Then, pixel-level permutation and bit-level permutation are used for more robust security of the cipher system. Finally, the diffusion process is used to change the pixels. Zahmoul et al. [21] introduced a construction of chaotic sequences by a chaotic map based on a Beta function. Then, the chaotic construction sequences are used in an encryption process that is included of permutation, diffusion and substitution operation that has high security. Huang et al. [22] proposed simple chaotic encryption of colour image by both plaintexts related permutation and diffusion to obtain high security and efficiency.

A remote sensing image encryption algorithm is introduced using AES by Zhang and Wang [23]. First, to reduce encryption times, the sender collects the values of 16 pixels together and converts them into large integers; second, the AES and messy chaotic will be used by the sender to encode large integers; finally, the encrypted image is from large, encrypted integers. Bas [24] proposed an improvement method called Chaotic Key Based RC6 (CKBRC6) which is used a Logistic map to generate round keys for the RC6 algorithm. Ramasamy et al. [25] proposed an image ciphering approach, which relies on a chaotic map and the generation of the symmetric key scheme. The approach uses block scrambling and adapted zigzag transformation, while key generation is established using the improved logistic-tent map. Confusion and diffusion are attained using pixel mixing. Hashim [26] introduced a method based on an enhanced blowfish algorithm and chaotic map. The image is scrambled and decomposed into several sunblock's randomly to dispatch the correlation of the pixels, and then each block is to an improved blowfish algorithm.

The main contributions of this paper are as follows:

- Proposing a 3D permutation model based on a hyperchaotic system.
- The proposed system has a block size of 2048 bits, supports the largest bit size and is practically unbreakable by brute force based on current computing power.
- Using key-dependent permutation
- Combining operations from different algebraic groups.

- One of the main characteristics of the proposed cipher is that it uses a symmetric sequence process that decrypts the ciphertext block based on previous ciphertext blocks. As a result, the presence of any single bit error in a ciphertext block affects the decryption process of all subsequent blocks because the decryption validity of the previous blocks is existing in the previous of adjacent ciphertext block.
- The proposed system used a large key space. The key size should be larger than 2^{100} to prevent attacks like brute force attack. The large keyspace is necessary to avoid a comprehensive search for a key.

3. Proposed Algorithm

For protecting the image content on the Internet, symmetric encryption for image encryption with the 3D permutation model based on a chaotic system is proposed. First, the chaotic sequences are generated by a hyperchaotic system besides two Chebyshev maps and then passed to the permutation model to generate the permuted image. Lastly, the proposed enhanced encryption is performed on permuted images. There are two categories pixel-level encryptions and bit-level encryptions. Both encryption approaches are significantly employed in the image permutation model and the proposed enhanced encryption in confusion and diffusion steps. Figure 1 illustrates the block diagram of the suggested algorithm of the image encryption section may be divided by subheadings. It should provide a concise and precise description of the experimental results, their interpretation, and the experimental conclusions that can be drawn.

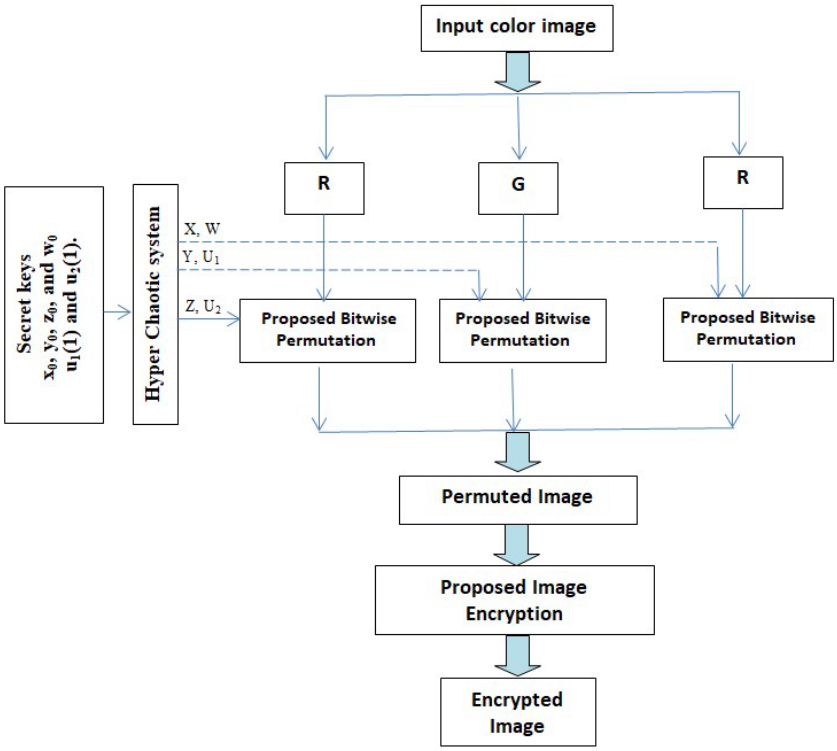


Fig. 1. Block diagram of RGB image encryption.

3.1. Secret key generation

In the design of scrambling, six random sequences are required. A four are generated by a hyperchaotic system besides two Chebyshev maps [8]. In our design of the scrambling method, we can use any chaotic system that generates six random chaotic sequences. The security is increased by using large keyspace and using key-dependent diffusion. These issues make the proposed method resists to differential, linear cryptanalysis and brute force attacks. This approach used a four-dimensional hyperchaotic system as well as four prime conditions and five system parameters, which Eq. (1) shows:

$$\begin{cases} {}^{(dx)}/_{dt} = a(y-x) + w \\ {}^{(dy)}/_{dt} = dx - xz - cy \\ {}^{(dz)}/_{dt} = xy - bz \\ {}^{(dw)}/_{dt} = yz - ew \end{cases} \tag{1}$$

The parameters are a, b, c, d and e. and the system is hyperchaotic if a = 35, b = 3, c = 12, d = 7 and e∈(0.085, 0.798), and it has two positive Lyapunov exponents, LE1 = 0.596, LE2 = 0.154. So the system is in a hyperchaotic state [8]. The attractor curves of the system are presented in Fig. 2. The two Chebyshev maps are modelled by Equation (2):

$$u\_1(i+1) = \cos\left(4 \times arcos(u\_1(i))\right)$$
$$u\_2(i+1) = \cos\left(4 \times arcos(u\_2(i))\right) \tag{2}$$

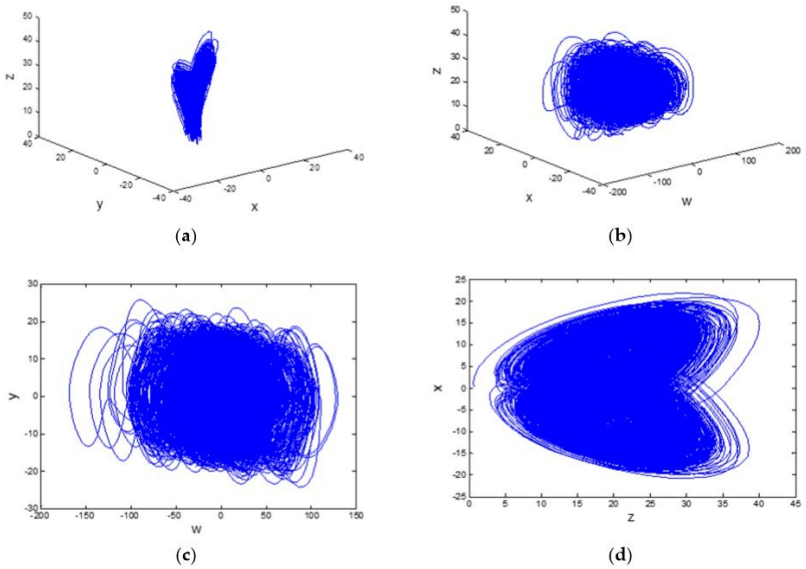


Fig. 2. The attractor of hyper-chaotic attractor.
(a) Plane of (x-y-z); (b) Plane of (w-x-z); (c) Plane of (w-y); (d) Plane (x-z).

The initial values are u1(1) and u2(1) [8]. The pseudorandom bit construction scheme includes the following steps:

Step 1**:** Determine the system parameters a, b, c, d, e beside the initial values x_0, y_0, z_0, and w_0 from Eq. (1) and the extra two values u1(1) and u2(1) from Eq.(2).

Step 2**:** After the highly chaotic system is repeated, four sequences are generated that are denoted as X = [x(i)], Y= [y(i)], Z = [z(i)] and W = [w(i)], respectively, where i = 1, 2, . . .

Step3: Repeated the two Chebyshev maps for constructing U1 and U2 sequences, respectively.

3.2. Proposed bitwise permutation

The proposed permutation approach is a key-dependent permutation that is employed a hyperchaotic system of four dimensions to generate secret keys (i.e., six chaotic sequences) as described in section (3.1). A high-dimensional chaotic map is more secure than a low-dimensional chaotic system, Therefore, they cannot resist the selected plaintext attack and the selection ciphertext attack [21] effectively.

The six subkeys are used to permute the rows and columns of the R, G, and B subbands of the input colour image *I* in a bit-level permutation. This permutation is performed to dispatch the correlation of the adjacent pixel. The block diagram of the proposed permutation-based chaotic is depicted in Fig. 3, and the permuted method is summarized in the algorithm (1).

Algorithm 1: Proposed Bitwise Permutation

Input:
I // Input colour image
r, c // row and column of the image I
Output
P // Permuted image
Step1: Decompose the input colour image *I* into three *r*, *g*, and *b* subbands.
Step2: The length of the chaotic sequences X, Y and Z, is l1 where l1= c×8 while the size of three sequences W, U1 and U2 is l2 where l2= r
Step3: Normalize the range of X, Y and Z from [0 1] to [1 c×8]
Step4: Normalize the range of W, U1 and U2 from [0 1] to [1 r]
Step5: Extend the value of red subband array *rbij* for i=1,..r,j=1,..c to a binary matrix: *ER*ij for i=1,..r,j=1,.. c×8.
Step6 Repeat step5 for green *gij* and blue *bij* subbands to generate binary sequences *EGij* and *EBij* for i=1,..r,j=1,.. c×8, respectively.
Step7: Permutated each row of the binary matrix *ER*ij with X to obtain a shuffled sequence of binary $Tij = \{Tij | i = 1, .., r, j = \cdots, c\times 8,\}$ such as following:

$$Tij = \text{Rotate}(ERij, xi),\ i = 1, \cdots, r,\ j = 1, \cdots, c\times 8$$

Step8: Permutated each column of the binary matrix ER with W to get a permuted binary sequence $Tij = \{Tij | i = 1, .., , c\times 8\ j = 1, \cdots, r$ such as following:

$$Tij = \text{Rotate}(ERji, wi),\ i = 1, \cdots, c\times 8,\ j = 1, \cdots r$$

Step9: Repeat step 7 for each row of the binary matrices EGij and EBij with Y and Z, respectively.
Step10: Repeat step 7 for each column of the binary matrices EGij and EBij with U1 and U2, respectively.
Step 11: Combined the permuted subbands in permuted image P.

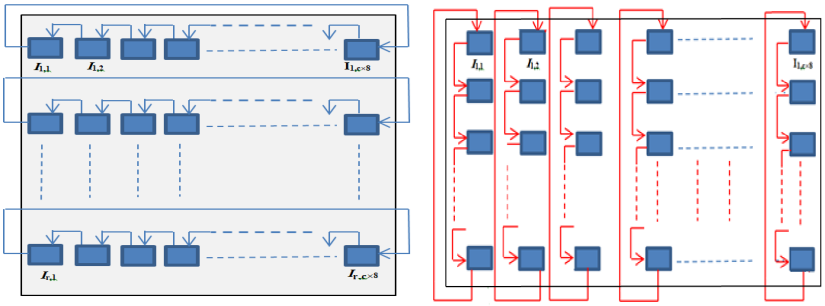
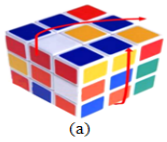


Fig. 3. Proposed bitwise permutation.
(a) Row bitwise permutation; (b) Column bitwise permutation.

Let the variables are initialized as follows:

l_1=32 (i.e., c×8)
X = 7, 10, 13, 20
Y = 3, 9, 15, 1
Z = 24,1,11,22
l_2=4
then
W = 1,0,2,2,1,1,2,0,0,3,1,2,3,3,2,1,2,2,0,0,3,3,1,0,0 ,1,0,2,2,0,1,2
U_1 = 3,1,2,0,0,3,2,1,0,2,1,3,0,3,2,1,2,2,0,0,3,3,1,1,3,1,1,1,2,1,1,2
U_2 = 1,0,2,3,1,3,1,2,0,2,1,3,0,3,2,1,1,2,0,2,3,3,1,1,0,1,3,1,2,1,2,1

As shown in Fig. 4, the two green and blue images have two columns and rows of zeros, while the encrypted images are wholly different from the original images.



(a)

17	127	244	99	222	192	123	0	100	63	0	15
128	245	63	127	127	125	80	0	16	70	0	100
115	15	0	226	63	0	3	0	0	12	0	0
222	31	0	15	192	0	116	0	0	199	0	0

(b)

Red	17	99	123	63	126	34	198	246	89	66	141	224
	128	127	80	70	17	160	31	212	114	224	94	118
	115	226	3	12	24	99	159	16	191	38	223	212
	222	15	116	199	247	76	125	224	20	41	55	144
Green	127	222	0	0	15	251	192	0	13	232	12	24
	245	153	0	0	0	122	204	128	14	33	12	128
	15	63	0	0	0	0	30	126	2	114	208	36
	31	192	0	0	15	224	0	0	1	218	194	66
Blue	244	192	100	15	192	100	15	244	84	20	8	160
	63	125	16	100	31	190	136	50	128	168	3	84
	0	0	0	0	0	0	0	0	10	96	128	16
	0	0	0	0	0	0	0	0	1	6	12	34

(c) (d) (e)

Fig. 4. Example of bitwise permutation (a) 3D image block
(b) 3D colour image values; (c) original RGB Decimal values;
(d) Row bitwise permutation; (e) Column bitwise permutation.

3.3. Proposed encryption method

An enhancement encryption algorithm is proposed. All traditional block cipher methods are not working correctly for image encryption. So the proposed method tries to enhance these ciphers to decorrelate pixel correlations of image.

The proposed algorithm is a 2048-bit RC6-like block cipher, and the key is 128 bits. The plaintext is divided into four parts bk_1, bk_2, bk_3 and bk_4, each of which is 512 bits. The *FB* (Function Block) function of the proposed system is used in a cascaded design instead of rounds, as it is shown in Fig. 5. The output is 2048 bits cbk_1, cbk_2, cbk_3 and cbk_4, each of which is 512 bits. The improved algorithm used a more complicated reversible mixing function as the permuted method. And this will further confuse the Feistel network entry values and guarantee a complete avalanche effect after the first two rounds. Algorithm (2) shows the steps of proposed image encryption.

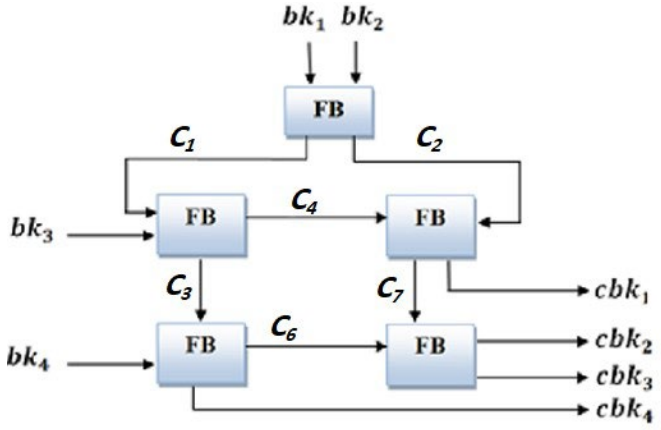


Fig. 5. Block diagram of proposed image encryption.

Algorithm 2: Proposed image encryption

Input:
P, // *Permuted image*
r, *c* // Rows and columns of the input image
Output:
E // Encrypted image
Step1: Divide the input image *P* into blocks bk_i *where* i=1,.., $r \times c$ each of size 512 bits.
Step2: For i=1 .. $r \times c$
// Each four blocks bk_i, bk_{i+1}, bk_{i+2}, bk_{i+3}
FB (bk_i, bk_{i+1}, C_1, C_2)
FB (bk_{i+2}, C_1, C_3, C_4)
FB (bk_{i+3}, C_3, cbk_{i+3}, C_6)
FB (C_2, C_4, C_7, cbk_i)
FB (C_6, C_7, cbk_{i+1}, cbk_{i+2})
I: =i+4
Store cbk_i, cbk_{i+1}, cbk_{i+2} *and* cbk_{i+3} in encrypted image *E*.
Step3: EndFor i

3.3.1. The FB function

The function FB of the proposed algorithm uses six of F sub-functions, representing one round of traditional RC6 algorithm [27] in type three Feistel network design. The input and output to F function is 128 bits and it required four subkeys Si, Si+1, Si+2, Si+3, Si+4, Si+5 where i=1,6,12,.., 174. The total required subkeys are 180. Each FB function is required 36 subkeys of 32 bits, as shown in Figs. 6 and 7. The type three Feistel network consists of dividing the input into four parts and applying a nonlinear function F represented by one round of RC6 to three parts. Consequently, the results of these parts are added; then, the resulting three parts are swapped with the fourth parts. The outputs of three nonlinear functions are input to the next one, which increases the propagation of local changes. Algorithm (4) presents the steps of function F.

In the FB design, the RC6 algorithm is used, which is one of the final list candidates (i.e, RC6, Twofish, Serpent, Mars and AES). Any one of them can be used in our design. The 5 FB are used to guarantee the numbers of rounds are greater than 20 and preserve the philosophical design of RC6. In designing the FB function a type three Feistel network is used; thus, we need 6 RC6 functions (i.e., 6 rounds) for each FB function. Hence the total numbers of rounds of the proposed encryption algorithm are 30 rounds.

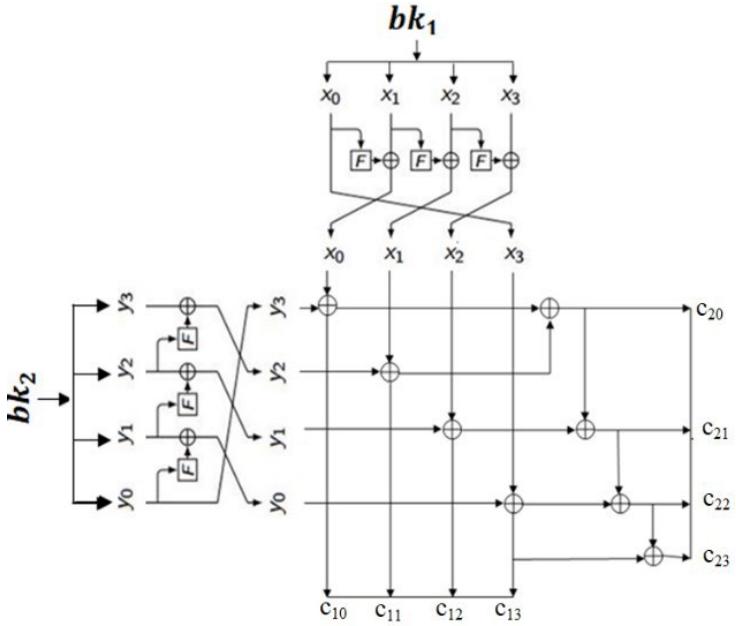


Fig. 6. The *FB* sub function of proposed image encryption.

Algorithm 3: FB function

Input:
bk_1, bk_2
Output:
Cbk_1, cbk_2
Step1: Divide bk_1 into four subblocks x_0, x_1, x_2, x_4 and bk_2 into four subblocks y_0, y_1, y_2 and y_3

Step2; $x_0 = F(x_0) \oplus x_1)$
$x_1 = F(x_1) \oplus x_2)$
$x_2 = F(x_2) \oplus x_3)$
$x_3 = x_0$
Step3: $y_0 = F(y_0) \oplus y_1)$
$y_1 = F(y_1) \oplus y_2)$
$y_2 = F(y_2) \oplus y_3)$
$y_3 = y_0$
Step4: $c_{10} = y_3 \oplus x_0$
$c_{11} = y_2 \oplus x_1$
$c_{12} = y_1 \oplus x_2$
$c_{13} = y_0 \oplus x_3$
Step5: $c_{20} = c_{10} \oplus c_{11}$
$c_{21} = c_{20} \oplus c_{12}$
$c_{22} = c_{21} \oplus c_{13}$
$c_{23} = c_{22} \oplus x_{13}$
Step6: Combine c_{10}, c_{11}, c_{12} and c_{13} into cbk_1
Step7: Combine c_{20}, c_{21}, c_{22} and c_{23} into cbk_2
Step8: The output is cbk_1, cbk_2

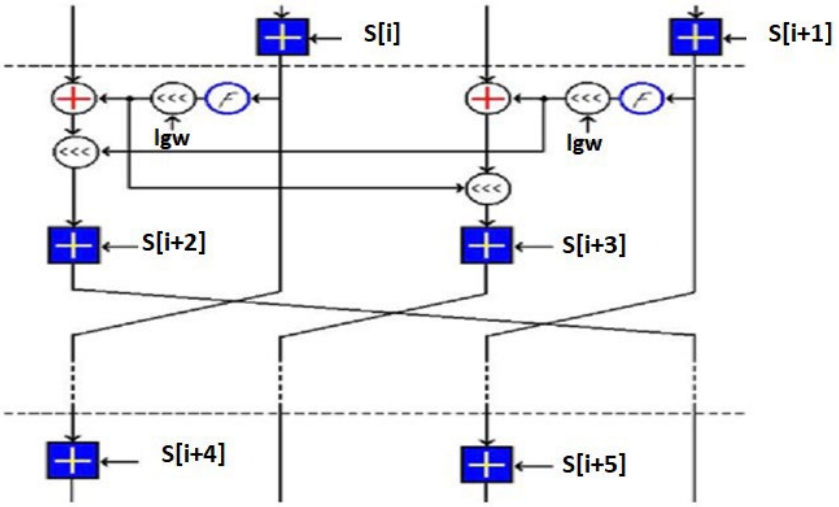


Fig. 7. The *F* subfunction of proposed image encryption.

Algorithm 4: F subfunction

Input
x, // 128 bits
S_i for i=1,..6
Output:
y, // 128 bits
Step1: Divided x into X, Y, M and N
Step2: $Y = Y + S_i$
Step3: $N = N + S_{i+1}$
Step4: $a = (Y \times (2Y + 1)) <<< lg(w)$

Step5: $b = (N \ x \ (2N + 1)) \ <<< \ \lg(w)$
Step6: $X = ((X \oplus a) \ <<< \ b) + S_{i+2}$
Step7: $M = ((M \oplus b) \ <<< \ a) + S_{i+3}$
Step8: (X, Y, M, N) = (Y, M, N, X)
Step9: $X = X + S_{i+4}$
Step10: $M = M + S_{i+5}$
Step11: Combine X, Y, M and N and into y

The sub function *F* of the proposed algorithm relays on its key, so this will avoid fixed output and increase the algorithm's non-linearity.

3.3.2. Key schedule of the proposed algorithm

The key schedule of the proposed algorithm and RC6 is practically identical. Indeed, the only difference is that for proposed, more words are required for encryption and decryption. The user gives a key of n bytes, where $0 \leq n \leq 255$. From this key, 180 words (w=32 bits each) are generated and stored in the array T[0, 180]. The generated subkeys are used in encryption/decryption. Algorithm 5 shows the steps of subkeys generation.

Algorithm 5: Subkeys generation

```
Input:
   n     // Key byte that is preloaded into c array L[0,1,2,…,r-1] of word.
   r     // Number of rounds
Output:
   T[0 , 180]    // ubkeys

1: T[0] = Pw    // PW= 0xB7E15163
2: For x = 1 to 180 do
     T[x] = T[x - 1] + Qw  //  QW= 0x9E3779B9
   End for i
     M= N = x= y = 0
     v=3 × max (r, 180)
  For s = 1 to v do
     M = T[x] = (T[x] + M + N) <<< 3
     N = D[x] = (D[x] + M + N) <<< (M + N)
      x = (x + 1) mod (180)
      y = (y + 1) mod r
   End for s
```

3.3.3. Design philosophy of proposed algorithm

To establish the requirements for the security, a block cipher must handle large input/output blocks, so the proposed algorithm increased size from 128 bit to 2048. And we have modified the design to use 32-bit registers rather than 64-bit registers; this has an advantage that is suitable for hardware design.

The philosophy of RC6 is to exploit operations (such as spin) that are executed efficiently on modern processors, and the proposed algorithm continues in this direction, taking advantage of the fact that 32-bit integer multiplication is now performed efficiently on most processors.

The number of rounds that enough to satisfy the required security for RC6 is 20 rounds with 44 subkeys, but the proposed system was employed the chaining mode in a cascade design. Thus the number of rounds is increased with more subkeys are used for each round.

RC6 studies failed to detect any weakness in the key generation. This provided one reason for using the same key generation in the proposed algorithm as RC6.

We believe that to attack the proposed algorithm, the greatest existing method for a cryptanalyst is to thoroughly search for a b-byte cipher key (or the extended key matrix S [0,.., 180] when the user-provided encryption key is particularly long). The user supplies a key of b bytes, where $0 \leq b \leq 255$. The work effort required for this is $\min\{2^{8b}; 2^{5760}\}$ operations, while the RC6 is $\min\{2^{8b}; 2^{1408}\}$.

4. Experimental Results

All of the experiments are accomplished on a desktop computer with Windows 10 Home edition operating system, Intel ® Core™ i7-2410M CPU 2.5GHZ Processor and 8GB RAM. Images such as 'A', 'Lena', 'Jellyfish',' Nike',' Peppers', 'Lichtenstein', and 'Baboon' are used as the original test images [28] (see Fig. 8). The set of parameters x(0), y(0), z(0), w(0)) = (0.377, 0.435, 0.799, 0.922 are used for encryption and decryption. Figure 9(a) is the original image, Fig. 9(b) is the permuted image, and the encrypted image is depicted in Fig. 9(c). As it noticed that the encrypted image is similar a noise image, and the original image has nothing to do with it. Thus, the encryption effect is good.

The RC6 algorithm is an excellent symmetric cipher and has a strong ability to counteract various types of attacks; Hence, it is widely used in data encryption. Today, many image encoders have considered RC6 reasonably secure but not suitable for image encoding. The reason is that the image has a large data size, and the data frequency is more than text data. Figure 10 shows the test images after they are encoded with the traditional RC6 algorithm and the proposed method.

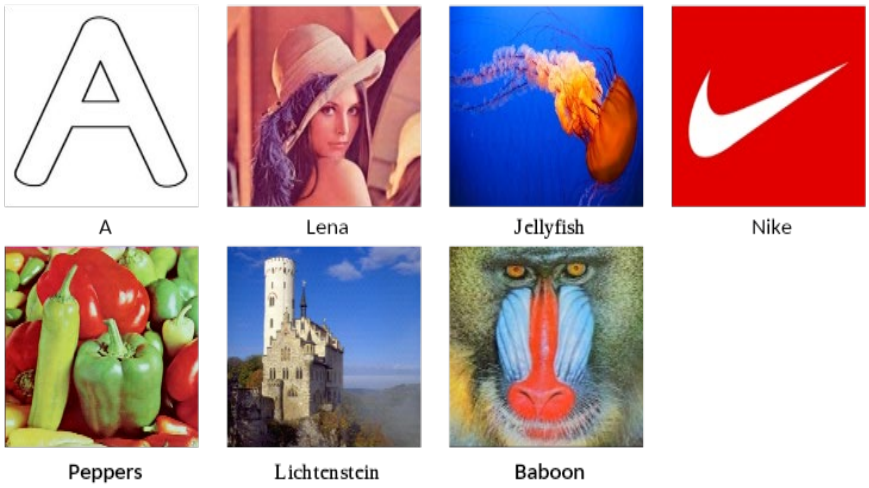


Fig. 8. The test images.

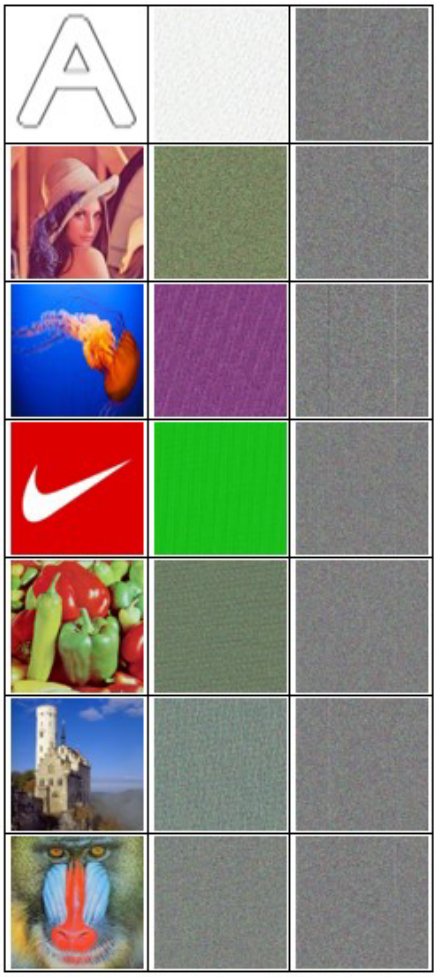


Fig. 9. Experiment results. (a) Original images; (b) Permuted image; (c) Encrypted image.

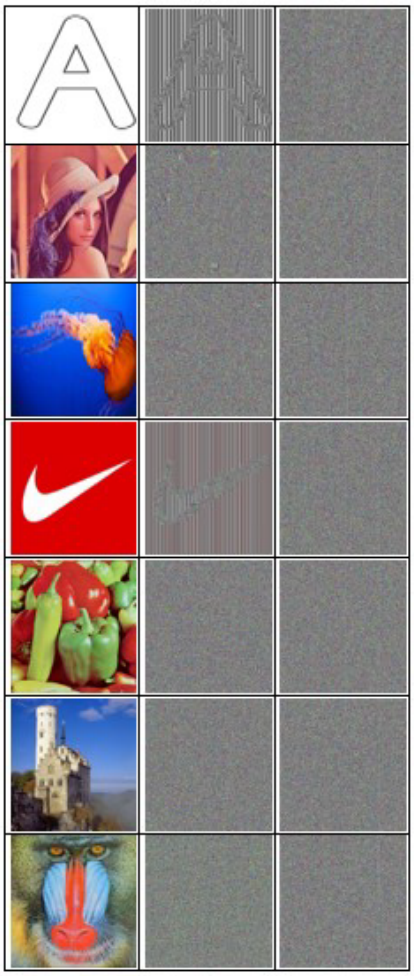


Fig. 10. Encrypted images. (a) Original image; (b) Encrypted image by RC6 algorithm; (c) Encrypted image by proposed algorithm.

4.1. Keyspace

The keyspace must be large enough for the desired encryption algorithm to resist a brute force attack. The keys of hyperchaotic system are represented by the four initial conditions of the chaotic system in Equation (1) and two values for the two Chebyshev maps in Equation (2). All initial conditions for two chaotic maps are precise to 10^{-15}, while the parameter e for e$\in$(0.085, 0.798) is precise to 10^{-12}, so the keyspace size is $(10^{15})^6 \times 10^{12} = 10^{102} \approx 2^{339}$. The size of the keyspace used in the proposed encryption based on RC6 is min$\{2^{8b}; 2^{5760}\}$ where $0 \leq b \leq 255$. The total keyspace for the proposed algorithm is $2^{339}+$ min$\{2^{8b}; 2^{5760}\}$ where $0 \leq b \leq 255$. Table 1 compares the size of the keyspace between the proposed algorithm and other related encryption algorithms. It is obvious that the keyspace size of the algorithm used is larger than most similar algorithms.

Table 1. keyspace comparisons.

Encryption Algorithm	Key Space
Zhu et al. [8]	2^{339}
Wang et al. [29]	2^{149}
Guesmi et al. [30]	2^{256}
Li et al. [31]	2^{299}
Li et al. [32]	2^{375}
Abdul-Adheem [33]	$\mathbf{2^{128}}$
Curiac et al. [34]	2^{357}
Proposed algorithm	$2^{339}+\min\{2^{8b}; 2^{5760}\}$ where $0 \leq b \leq 255$.

Avalanche effect property is computed to show key sensitivity and to verify the strength of the proposed system against differential cryptanalysis. When the slight change is taken in the initial condition of the chaotic system, the change in the number of bits in encryption is around 50%. This shows that the proposed system has high confusion, high sensitivity to the key, and all bits of the key contribute uniformly and significantly to the encrypted bits. Figure 11 depicts the Avalanche effect for encrypted 256 × 256 Lena image.

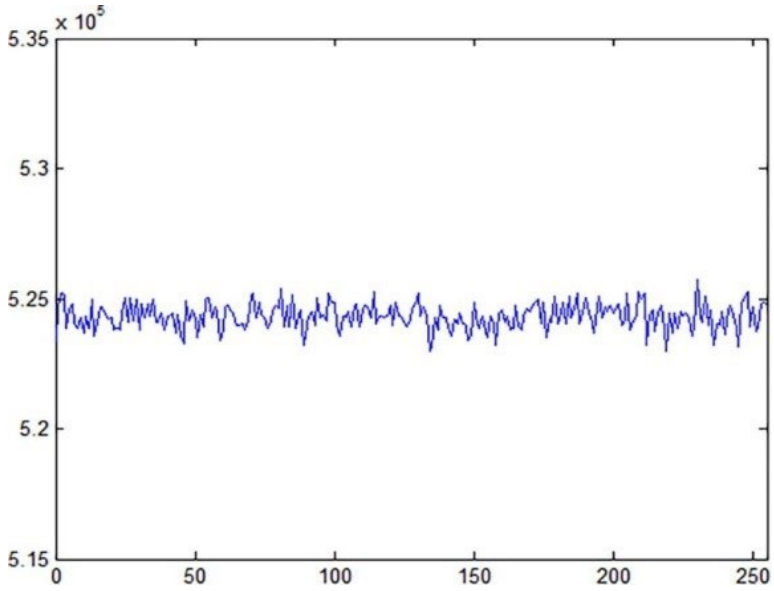


Fig. 11. Avalanche effect property - the number of bits changed in the ciphered Lena with slight changes on the initial condition.

Figure 12 shows the number of bits changed in the encrypted image with a change of one bit per pixel in the original "Lena" of 256 × 256 images. It is noticed that there is approximately 50% change in the cipher accomplished by one change in pixel of the original image. This shows that all bit positions of pixels in the original image contribute significantly to the cipher bits without any biases. This indicates a strong diffusion feature that the proposed system possesses.

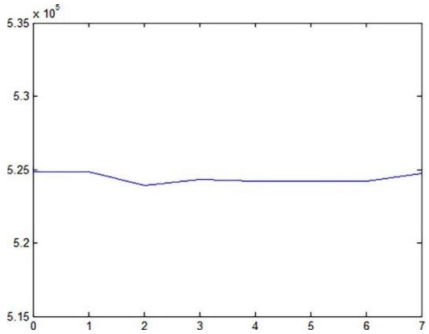


Fig. 12. Avalanche effect property - the number of bits changed in the ciphered "Lena" image with one bit per pixel change in each position 8.

4.2. Security analysis

The proposed algorithm shows its effective resistance from statistical attack, as the RGB colour image will be used to analyse the security of the proposed encryption system.

4.2.1. Histogram analysis

The histogram reflects the exact representation of pixel value distribution. To obtain the desired encoding results, the histogram of the ciphering image is permanently uniform [33]. Figure 13(a) are the original test images, Fig. 13(b) are the histograms of test images, Fig. 13(c) are the histograms permuted images, Fig. 13(d) are the histograms encrypted images by the proposed algorithm.

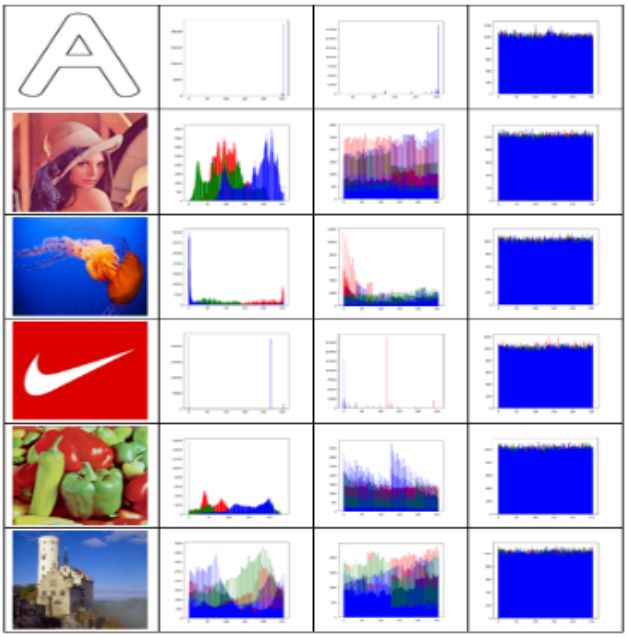


Fig. 13. Histogram of images. (a) Original images; (b) Histogram of original images; (c) Histogram of permuted images; (d) Histogram of Encrypted image by the proposed algorithm.

All the histograms of Figure 13(d) show that the obtained results are uniformly distributed, and quite different from the Air-eld results. Therefore, the statistical attack can be effectively resisted by the proposed algorithm.

4.2.2. Correlation analysis

To increase the encryption strength and security of the encrypted image, the correlations between adjacent pixels in the original image must be very few. Thus, encryption should have a significant impact because the bulk of the original image area has very close grayscale values. As a test, the vertical, horizontal, and diagonal directions will be used to calculate the correlation coefficients for all adjacent pixels, using equation (3).

$$r_{xy} = \frac{cov(x.y)}{\sqrt{D(x)\times\sqrt{D(y)}}} \tag{3}$$

where

$$cov(x,y) = \frac{1}{N}\sum_0^N (x_i - E(x))) \ ,$$

$$D(x) = \frac{1}{N}\sum_0^N x_i - E(x)^2 \ ,$$

$$E(x) = \frac{1}{N}x_i$$

The two adjacent pixel values from four directions are represented by x, y, and N is the image pixels' number. Table 2 lists the correlation coefficients of the original and their encoded images, as well as the vertical directions (V), horizontal (H), diagonal (D) directions. Table 3 shows that the proposed algorithm can withstand hacker attacks according to the value of the correlation coefficient, where it is close to 0 for the encrypted image, while for the plain image it is close to 1. Figure 14 shows the vertical, horizontal and diagonal directions for image 'Lena', 'A' and 'Nike' images.

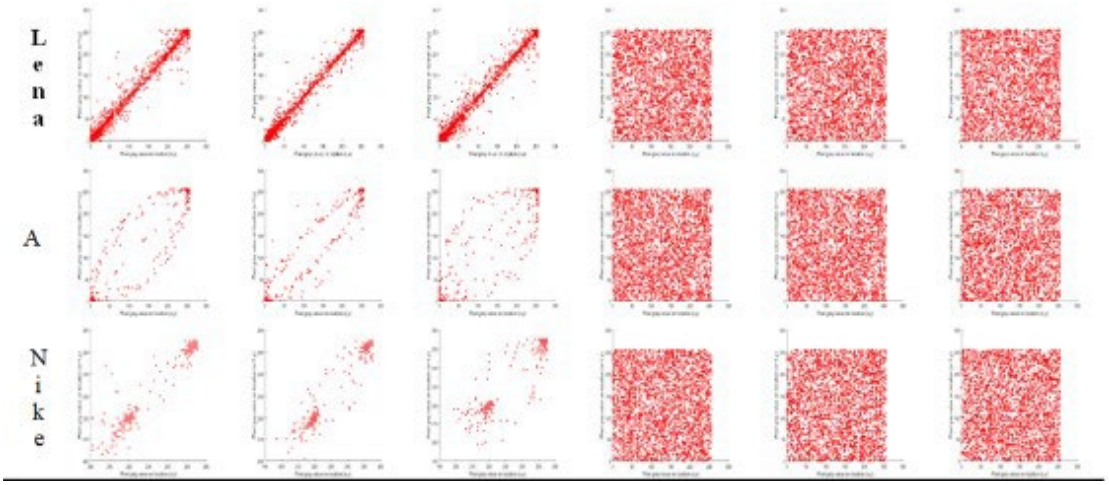


Fig. 14. Correlation coefficients of Lena, A and Nike images and the corresponding encrypted image in R, G, and B channels.

Table 2. The correlation coefficients of original images and their encrypted images.

Image	Colour	Plain-Image			Cipher-Image		
		Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
Lena	Red	0.9887	0.9947	0.9815	-0.0023	0.0069	-0.0014
	Green	0.9892	0.9950	0.9826	-0.0049	0.0074	0.0052
	Blue	0.9793	0.9898	0.9668	-0.0040	0.0086	-0.0033
A	Red	0.9493	0.9656	0.9185	-0.0013	0.0051	-0.0030
	Green	0.9493	0.9656	0.9185	-0.0005	0.0090	-0.0008
	Blue	0.9493	0.9656	0.9185	0.0027	0.0086	-0.0015
Nike	Red	0.9900	0.0083	0.9744	-0.0037	0.0071	-0.0027
	Green	0.9971	0.9966	0.9909	-0.0029	0.0073	-0.0028
	Blue	0.9971	0.9966	0.9910	-0.0025	0.0082	-0.0026
Jellyfish	Red	0.9928	0.9953	0.9920	-0.0066	0.0087	-0.0032
	Green	0.9764	0.9838	0.9748	-0.0039	0.0112	-0.0029
	Blue	0.9928	0.9953	0.9920	-0.0040	0.0067	-0.0065
Baboon	Red	0.9866	0.9820	0.9712	-0.0038	0.0073	-0.0024
	Green	0.9780	0.9711	0.9533	-0.0036	0.0084	-0.0056
	Blue	0.9873	0.9843	0.9738	-0.0015	0.0046	-0.0072
Peppers	Red	0.9613	0.9641	0.9547	-0.0029	0.0089	-0.0025
	Green	0.9804	0.9811	0.9681	-0.0050	0.0080	-0.0048
	Blue	0.9645	0.9643	0.9462	-0.0019	0.0049	,-0.0035
Lichten-stein	Red	0.9626	0.9768	0.9486	-0.0038	0.0073	-0.0020
	Green	0.9472	0.9675	0.9276	-0.0036	0.0084	-0.0037
	Blue	0.9731	0.9825	0.9629	-0.0015	0.0046	-0.0033

4.2.3. Entropy analysis

Entropy is defined as a measure of the randomness of the content information in an image, which means minimum average coding length in bits per pixel that can be achieved with an ideal coding scheme without any information loss. The higher of entropy value, the more confusion the image information [35, 36]. For discrete 2D images, the information entropy E is calculated using Equation (4):

$$E = -\sum_{i=0}^{n} P_i log_2(P_i) \quad (4)$$

where pi represents a probability of grey value, where the probability of the encryption image will be equally distributed, that is, the probability of each pixel value of [0, 255] is 1/256, and the maximum entropy is 8 bits [37].

The information entropy of our original test images is listed on Table 3, with the information entropy of the encrypted. The entropies of the encrypted images by the proposed algorithm approximately near to 8 bits. The entropy and correlation values obtained in the proposed method for a 256 x 256 "Lena" image will be compared with other methods as shown in Table 4.

Table 3. The entropy of original images and their encrypted images.

Image	Original			Permutation			Encryption		
	R	G	B	R	G	B	R	G	B
A	1.1608	1.1608	1.1608	3.2945	3.3032	3.3053	7.9978	7.9981	7.9980
Lena	7.2865	7.5592	7.0527	7.9939	7.9821	7.9952	7.9980	7.9980	7.9981
Jellyfish	5.9170	7.2944	7.2865	7.3028	7.9925	7.9516	7.9982	7.9982	7.9981
Nike	1.1818	1.0344	1.0159	4.4702	4.3242	4.4986	7.9979	7.9978	7.9978
Peppers	7.3444	7.5501	7.1173	7.9955	7.9786	7.9527	7.9981	7.9981	7.9980
Lichten s-tein	7.5346	7.4783	7.4960	7.9729	7.9949	7.9859	7.9980	7.9977	7.9981
Baboon	7.6202	7.3139	7.6277	7.9957	7.9976	7.9976	7.9980	7.9979	7.9978

Table 4. Comparison of performance with other methods

Measure	[38]	[39]	[40]	[41]	[42]	[43]	[44]	Proposed
Horizontal correlation	0.0327	0.9407	0.0018	−0.0230	0.0020	−0.0067	−0.0098	−0.0031
Vertical correlation	0.0219	−0.027	0.0011	0.0019	−0.0007	−0.0137	−0.0050	0.0072
Diagonal Correlation	0.0180	−0.014	−0.0012	−0.0034	−0.0014	−0.0563	−0.0013	−0.0025
Entropy	7.9993	n/a	7.9994	7.9974	7.9970	n/a	7.9974	7.9981

4.2.4. NPCR and UACI analysis

Standard metrics such as NPCR (Pixel Change Rate) and UACI (Unified Average Change Density) are used to exam the variation between the original image P1 and the ciphered image C1. Then, if D is matrixes in the same size image as P1 and C1, D (i, j) is defined as follows [45]:

$$D_{ij} = \begin{cases} 1 & P_i(i,j) \neq C_1(i,j) \\ 0 & else \end{cases} \quad (5)$$

$$NCPR = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \frac{D_{ij}}{M \times N} \times 100\% \quad (6)$$

By applying the proposed system, it was conducted 100 tests. It achieved NPCR metric between 98.8 and 99.33 and between 32.17 and 33.18 UACI test values, which is very close to the theoretical maximum, confirming that the proposed system will be immune to the differential attacks.

$$UACI = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \frac{|P_1(i,j) - C_1(i,j)|}{255} \times 100\% \quad (7)$$

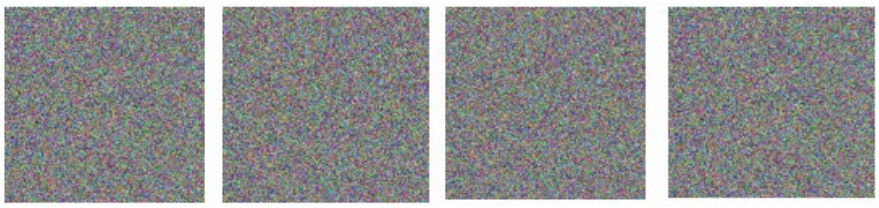
Table 5 shows the NPCR and UACI of encrypted images by the proposed system.

Table 5. NPCR and UACI of encrypted images by the proposed system

Image	NCPR			UACI		
	R	G	B	R	G	B
A	98.91	98.92	98.96	32.34	32.33	32.29
Lena	99.21	99.30	99.11	33.38	33.41	33.49
Jellyfish	98.98	98.94	98.91	32.31	32.28	32.29
Nike	98.45	98.99	98.77	32.12	32.15	32.11
Peppers	99.28	99.31	99.29	33.30	33.32	33.31
Lichtenstein	99.11	99.23	99.22	33.48	33.51	33.49
Baboon	99.33	99.30	99.23	33.51	33.55	33.53

4.2.5. Key sensitivity

In this test, we made a slight change in the initial condition to confirm key sensitivity. Figure 15 shows the ciphered image of 'Lena' with different keys with a slight change. We notice that the ciphered images wholly differ from the original image.



(a) (b) (c) (d)

Fig. 15. Ciphered image with the wrong keys. (a) Image encryption with a slight change in x_0 by 10^{-4}; (b) Image ciphered with a slight change in y_0 by 10^{-4}; (c) Image ciphered with slight change in z_0 by 10^{-4}; (d) Image ciphered slight change in w_0 by 10^{-4}.

5. Conclusion

The proposed scheme is an image-enhanced coding, which includes two main components: pixel scrambling and pixel coding. The algorithm has the following advantages, which can be generated from the experimental results of theoretical analysis: The key space is large. Thus, the global key search and the matching ciphertext attack is not possible, the pixel distribution of the encrypted image is consistent, the correlation between adjacent pixels of the ciphertext is very low, and the ciphertext images have entropy information that is very close to the ideal value of 8, the mixing operations are used from different algebraic group: XOR, addition, rotation and multiplication. Thus, the suggested algorithm has perfect opportunities for application in secure image communication and storage applications. The results of the avalanche effect characteristics show that the strength provided by the probabilistic approach against differential and statistical cryptanalysis.

References

1. Zhang, W.; Zhu, Z. ; and Yu, H. (2019). A symmetric image encryption algorithm based on a coupled logistic-bernoulli map and cellular automata diffusion strategy. *Entropy*, 21(5): 504, 1-23.
2. Aboughalia, R.; Alkishriwo, O.; and Alkishriwo, S. (2018). Colour image encryption based on chaotic block permutation and XOR operation. *Libyan International Conference on Electrical Engineering and Technologies* (*LICEET*2018), Libya, 492-497.
3. Matthews, R. (1989). On the derivation of a chaotic encryption algorithm. *Cryptologia*, 13(1), 29-42.
4. Baptista, M.S. (1998). Cryptography with chaos. *Physics Letters A*, 240(1-2), 50-54.
5. Zhu, H.; Zhang, X.; Yu, Ha; Zhao, C.; and Zhu, Z. (2016). A novel image encryption scheme using the composite discrete chaotic system. *Entropy*, 18(8), 276, 1-27.
6. Wu, X.; Li, Y.; and Kurths, J. (2015). A new colour image encryption scheme using CML and a fractional-order chaotic system. *PLOS ONE*, 10(3): e0119660.
7. Fridrich, J. (1998). Symmetric ciphers based on two-dimensional chaotic maps. *International Journal of Bifurcation and Chaos*, 8(6), 1259-1284.

8. Zhu, S.; Zhu, C.; and Wang, W. (2018). New image encryption algorithm based on chaos and secure Hash SHA-256. *Entropy*, 20(9): 716, 1-18.
9. Younes, M.A.B.; and Jantan, A. (2008). Image encryption using block-based transformation algorithm. *IAENG International Journal of Computer Science*, 35(1), 15-23.
10. Zhang, G.; and Liu, Q. (2011). A novel image encryption method based on total shuffling scheme. *Optics Communications*, 284(12), 2775-2780.
11. Liu, H.; and Wang, X. (2011). Colour image encryption using spatial bit-level permutation and high-dimension chaotic system. *Optics Communications*, 284(16-17), 3895-3903.
12. Boriga, R.E.; Dascalescu, A.C.; and DiaconuI, A.-V. (2014). A new fast image encryption scheme based on 2d chaotic maps. *IAENG International Journal of Computer Science*, 41(4), 249-258.
13. Wang, X.; Teng, L.; and Qin, X. (2012). A novel colour image encryption algorithm based on chaos. *Signal Processing*, 92(4), 1101-1108.
14. Wang, X.-Y.; Zhang, Y.-Q.; and Bao, X.-M. (2015). Colour image encryption scheme using permutation-substitution based on chaos. *Entropy*, 17(6), 3877-3897.
15. Murillo-Escobar, M.A.; Cruz-Hernandez, C.; Abundiz-Perez, F.; Lopez-Gutierrez, R.M.; and del Campo, O.A. (2015). A RGB image encryption algorithm based on total plain image characteristics and chaos. *Signal Process*, 109, 119-131.
16. Zheng, Y.; and Jin, J. (2014). A novel image encryption scheme based on Henon map and compound spatiotemporal chaos. *Multimedia Tools and Appl*ications, 74, 7803-7820.
17. Ye, G. (2013). Image scrambling encryption algorithm of pixel bit based on chaos map. *Pattern Recognition Letters*, 31(5), 347-354.
18. Fu, C.; Lin, B.-b.; Miao, Y.-s.; Liu, X.; and Chen, J.-j. (2011). A novel chaos-based bit-level permutation scheme for digital image encryption. *Optics Communications*, 284(23), 5415-5423.
19. Li, C.; Lin, D.; and Lü, J. (2017). Cryptanalyzing an image-scrambling encryption algorithm of pixel bits. *IEEE MultiMedia*, 24(3), 64-71.
20. Li, Y.; Wang, C.; Chen H. (2017). A hyper-chaos-based image encryption algorithm using pixel-level permutation and bit-level permutation. *Optics and Lasers in Engineering*, 90, 238-246.
21. Zahmoul, R.; Ejbali, R.; and Zaied, M. (2017). Image encryption based on new Beta chaotic maps. *Optics and Lasers in Engineering*, 96, 39-49.
22. Huang, L.; Cai, S.; Xiao, M.; and Xiong, X. (2018). A simple chaotic map-based image encryption system using both plaintext related permutation and diffusion. *Entropy*, 20(7): 535, 1-20.
23. Zhang, X.; and Wang, X. (2018). Remote-sensing image encryption algorithm using the advanced encryption standard. *Applied Sciences*, 8(9), 1540.
24. Bas, M. (2018). Digital image encryption using logistic chaotic key-based RC6. *International Journal of Computer Applications*, 182(2), 17-23.
25. Ramasamy, P.; Ranganathan, V.; Kadry, S.; Damaševicius, R.; and Blažauskas, T. (2019). An image encryption scheme based on block

scrambling, modified zigzag transformation and key generation using enhanced logistic - tent map. *Entropy*, 21(7): 656, 2-17.

26. Hashim, A.T.; Jassem, A.H.; and Ali, S.A. (2021). A novel design of Blowfish algorithm for image security. *Journal of Physics Conference Series*, 1818(1): 012085.
27. Rivest, R.L. (1995). *The RC*5 *encryption algorithm.* In: B. Preneel (ed.) *Fast software encryption*. Springer Berlin Heidelberg, Berlin, Heidelberg 1995, 86-96.
28. Retrieved October 5, 2020, from https://homepages.cae.wisc.edu/~ece533/ images.
29. Wang, X.; Zhu, X.; Wu, X.; and Zhang, Y. (2017). Image encryption algorithm based on multiple mixed hash functions and cyclic shift. *Optics and Lasers in Engineering*, 107, 370-379.
30. Guesmi, R.; Farah, M.A.B.; Kachouri, A.; and Samet, M. (2016). A novel chaos-based image encryption using DNA sequence operation and secure hash algorithm sha-2. *Nonlinear Dynamics*, 83, 1123-1136.
31. Li, S.; Chen, G.; and Mou, X. (2005). On the dynamical degradation of digital piecewise linear chaotic maps. *International Journal of Bifurcation and Chaos*, 15(10), 3119-3151.
32. Li, S.; Chen, G.; Wong, K.-W.; Mou, X.; and Cai, Y. (2004). Baptista-type chaotic cryptosystems: Problems and countermeasures. *Physics Letters A*, 332(5-6), 368-375.
33. Abdul-Adheem, W.R. (2020). Enhancement of magnetic resonance images through piecewise linear histogram equalization. (*JESTEC*), 15(3), 2023-2039.
34. Curiac, D.I.; Iercan, D.; Dranga, O.; Dragan, F.; and Banias, O. (2007). Chaos-based cryptography: End of the road?. *In Proceedings of the International Conference on Emerging Security Information*, *System and Technologies*, Valencia, Spain, 14-20 October, 71-76.
35. Chai, X.; Chen, Y.; and Broyde, L. (2017). A novel chaos-based image encryption algorithm using DNA sequence operations. *Optics and Lasers in Engineering*, 88, 197-213.
36. Taha, D.B.; Taha, T.B.; Al Dabagh, NB..; Ngadiran, R.; and Ehkan, P. (2019). Digital image watermarking algorithm based on texture masking model. *Journal of Engineering Science and Technology* (*JESTEC*), 14(6), 3347-3360.
37. Hashim, A.T.; and Jalil, B.D. (2020). Color image encryption based on chaotic shift keying with lossless compression. *International Journal of Electrical and Computer Engineering* (*IJECE*), 10(6) 5736~5748.
38. Li, Y.; Li, X.; Jin, X.; Zhao, G.; Ge, S.; Tian, Y.; Zhang, X.; Zhang, K.; and Wang, Z. (2015). An image encryption algorithm based on zigzag transformation and 3-dimension chaotic logistic map. *In Applications and Techniques in Information Security*; *Springer: Berlin*/*Heidelberg*, Germany, 3-13.
39. Ahmad, J.; and Hwang, S.O. (2016). A secure image encryption scheme based on chaotic maps and affine transformation. *Multimedia Tools and Appl*ications, 75, 13951-13976.
40. Zhang, Y.; and Xiao, D. (2014). An image encryption scheme based on rotation matrix bit-level permutation and block diffusion. *Communications in Nonlinear Science and Numerical Simulation*, 19(1), 74-82.

41. Xu, L.; Li, Z.; Li, J.; and Hua, W. (2016). A novel bit-level image encryption algorithm based on chaotic maps. *Optics and Lasers in Engineering*, 78, 17-25.
42. Wang, X.-Y.; Zhang, Y.-Q.; and Bao, X.-M. (2015). A novel chaotic image encryption scheme using DNA sequence operations. *Optics and Lasers in Engineering*, 73, 53-61.
43. Hussain, I.; Shah, T.; and Gondal, M.A. (2012). Image encryption algorithm based on PGL (2, GF (28)) S-boxes and TD-ERCS chaotic sequence. *Nonlinear Dynamics*, 70, 181-187.
44. Wang., X.; and Zhang, H.-L. (2015). A colour image encryption with heterogeneous bit-permutation and correlated chaos. *Optics Communications*, 342, 51-60.
45. Hashim, A.T.; Hasan, A.M.; and Abbas, H.M. (2020). Design and implementation of proposed 320 bit RC6-cascaded encryption/decryption cores on altera FPGA. *International Journal of Electrical and Computer Engineering* (*IJECE*), 10(6), 6370-6379.