A new four-dimensional hyper-chaotic system for image encryption

Huda R. Shakir, Sadiq A. Mehdi, Anwar A. Hattab
Department of Computer Science, Mustansiriyah University, Baghdad, Iraq

Article Info

ABSTRACT

Currently, images are very important with the rapid growth of communication networks. Therefore, image encryption is a process to provide security for private information and prevent unwanted access to sensitive data by unauthorized individuals. Chaos systems provide an important role for key generation, with high randomization properties and accurate performance. In this study, a new four-dimensional hyper-chaotic system has been suggested that is used in the keys generation, which are utilized in the image encryption process to achieve permutation and substitution operations. Firstly, color bands are permuted using the index of the chaotic sequences to remove the high correlation among neighboring pixels. Secondly, dynamic S-boxes achieve the principle of substitution, which are utilized to diffuse the pixel values of the color image. The efficiency of the proposed method is tested by the key space, histogram, and so on. Security analysis shows that the proposed method for encrypting images is secure and resistant to different attacks. It contains a big key space of (2^627) and a high sensitivity to a slight change in the secret key, a fairly uniform histogram, and entropy values nearby to the best value of 8. Moreover, it consumes a very short time for encryption and decryption.

Keywords: Hyper-chaotic system Image encryption Permutation Substitution

1. INTRODUCTION

With the rapid advancement of digital technologies and communication networks, multimedia communication such as images, videos, and audio is becoming increasingly significant and indispensable in today’s society [1]–[3]. However, network openness and sharing introduce several security risks to digital communication. Digital images, as a significant medium of multimedia communication, play a major part in medical, biological, military, and social life. Digital images contain a lot of important information, so the security of image data attracts more and more attention [4], [5].

Due to the strong correlation between image pixels, high redundancy, and large data capacity [6], [7], therefore, conventional encryption algorithms like advanced encryption standard (AES), data encryption standard (DES), and Rivest-Shamir-Adleman (RSA) algorithms are not appropriate for image encryption [8], [9]. As a result, many researchers have proposed new image coding algorithms that satisfy the criteria of confusion and diffusion [10]. Compared with conventional encryption systems, chaotic systems have stronger advantages such as sensitivity to initial value, ergodicity, randomness, and other characteristics [11]–[13], which provide a quick and secure way to protect data transmitted over communication channels like the internet. In order to ensure the effectiveness of the algorithms, Fredrich was the first to combine the chaotic system with cryptography theory in 1998 [10], [14].

Corresponding Author:
Huda R. Shakir
Department of Computer Science, University of Mustansiriyah
Alkadhimiya Street, Baghdad, Iraq
Email: hudarashid@uomustansiriyah.edu.iq

This is an open access article under the CC BY-SA license.
Chaotic systems have been used in many studies to encrypt images. Below is a review of some relevant work. Ali and Ali [15] proposed a new approach for encrypting color images using chaotic maps. Three stages are involved in the construction of the cipher image. The first phase involves permuting the digital image using a chaotic map. The second stage employs a chaotic substitution box to perform pixel substitutions, and the third stage employs the Boolean operation exclusive OR (XOR) to blend chaotic logistic-based random sequences. The red, green and blue (RGB) components of an image that have been scrambled using permutation, substitution, and XOR operations have good security and resistance to various types of encryption attacks.

Teng et al. [16] proposed a new image encryption method based on a cross 2D hyper-chaotic map. The keys are constructed by the hash function (SHA-512) and the plain image. Firstly, the color image is transformed to a merged bit-level array and scrambled using the column and row cogenesis shift process. The scrambled matrix is then diffused using a choosing sequence that is dependent on the chaotic sequence. Lastly, the diffusion matrix is decomposed in order to obtain the color image encryption. Simulations and security analysis show that the method works well at encrypting the color image and provides adequate protection against various types of attacks.

Tariq et al. [17], suggested an image encryption scheme based on Lorenz chaotic systems and analyzed the flaws of the public key encryption method based on logarithm and provided an additional layer of security to prevent it from cryptographic assaults. According to both security analyses and experimental findings, the suggested encryption method has proven to be resistant to both linear and differential attacks. Patro et al. [18], suggested an image encryption approach based on multiple levels of permutations and diffusions. Bit-level permutations are done with a 4-D hyper-chaotic map. The diffusion step is done with the piecewise linear chaotic map (PWLCM) system, and block-level permutations are done with Alpar's map. According to simulation and security analysis, the comparison tests show that the proposed method is secure and immune to most common threats.

Talhaoui et al. [19], suggested a new image encryption method dependent on the chaotic map of the Bulban. The new scheme’s security is provided via a substitution and permutation network, which uses a rotating shift of columns and rows to remove the high correlation among neighboring pixels. Then, the pixels values are masked the of with XOR and the modulo function to prevent any data from leaking into the system. Tests and simulations have shown that the system is safe and very fast for real-time images.

Most encryption algorithms have three key problems: insufficient security, limited processing capacity, and poor encryption efficiency. So, traditional algorithms are not suitable for image encryption, in addition to the problems of low-chaotic systems such as small key space and low complexity. To overcome these problems, in this paper, a new color image encryption method built on a novel 4-D chaotic system has been suggested to offer higher security and randomness. Initially, the position of pixels are scrambled through the sequences chaotic; then, high confusion is provided by the dynamic S-Boxes constructed using DNA computing with the new chaotic system. An experimental and security analysis was performed on the proposed image encryption algorithm to verify the security and efficiency of the encryption.

2. THEORETICAL BACKGROUND

2.1. The new chaotic system

The novel 4D-four-dimensional hyperchaotic system with ten positive parameters is proposed. It has more complex properties than lower-dimensional chaotic systems. An entirely new 4D autonomous system can be obtained by the following differential equation in system (1).

\[
\begin{align*}
\frac{dx}{dt} &= -ax - bw + cyz + ze^y \\
\frac{dy}{dt} &= dy + ex - fxz - xe^z \\
\frac{dz}{dt} &= -gz + hxy \\
\frac{dw}{dt} &= -bw + i x z + jyz
\end{align*}
\]

The states of system are x, y, z, and w, t \in \text{real number} and the positive parameters are \(b, d, a, c, g, e, h, f, j, i\) which displays a chaotic attractors in the new 4-D chaotic system by the initial conditions of: \(x_0 = 0.2, y_0 = 0.4, z_0 = 1.5,\) and \(w_0 = 0.8\) and parameter values of: \(a = 3.1, b = 2.1, c = 15.8, d = 1.1, e = 16.5, f = 1.5, g = 2.4, i = 5.1, h = 26.6,\) and \(j = 12.9.\) Figure 1(a) shows the chaotic attractors in (y-x-w), Figure 1(b) in (x-y-z), Figure 1(c) in (x-z-w), and Figure 1(d) in (y-x-z). The four Lyapnov Exponents of system (1), by parameters are obtained: \(LE_1=4.05761, LE_2=0.347562, LE_3=-3.94257\) and \(LE_4=-6.61896.\) Due to LE1 and LE2 are positive Lyapnon exponents. Therefore, the new system is a hyper chaotic.

A new four-dimensional hyper-chaotic system for image encryption (Huda R. Shakir)
2.2. Waveform analysis of the novel chaotic system

The waveform of a chaotic system should be aperiodic in order to show that the suggested system is a chaotic system, as shown in Figure 2. Figure 2(a) depicts the $x(t)$ waveform, Figure 2(b) the $y(t)$ waveform, Figure 2(c) the $z(t)$ waveform, and Figure 2(d) the $w(t)$ waveform in the time domain. By plotting the results of a MATHEMATICA simulation, it is clear that the waveform of the system has complex behavior, chaotic motion, and non-periodic characteristics. So, the time-domain waveform has non-cyclical characteristics so that it can tell the difference between several periodic motions with complicated behaviors and chaotic motion.

2.3. The sensitivity to initial conditions of the novel chaotic system

Long-term unpredictability is perhaps the most key characteristic of a chaotic system. This is due to the sensitivity of solution dependency on initial variables. Two distinct initial states, despite how close they begin, will eventually separate widely. As a result, for every initial condition with a finite number of digits of accuracy, a future time will come when no reliable predictions can be formed about the state of a system. Figures 3(a) to 3(d) demonstrate the sensitivity of the chaotic trajectories’ evolution to the initial conditions in $x(t)$, $y(t)$, $z(t)$, and $w(t)$. For the solid line, the initial values are $x_0=0.2$, $y_0=0.4$, $z_0=1.5$, and $w_0=0.8$; for the dashed line, they are $x(0)=0.2$, $y(0)=0.400000000000001$, $z(0)=1.5$, and $w(0)=0.8$.
A new four-dimensional hyper-chaotic system for image encryption (Huda R. Shakir)
sequences ($x_i$, $y_i$, $z_i$, and $w_i$), which have been transformed to vectors $k_1$, $k_2$, $k_3$, and $k_4$ from chaotic sequences, also, using algebraic DNA processes (exclusive-or, subtraction, and addition), which are used to make a set of S-boxes (1616), each chaotic sequence is turned into a binary sequence and then changed to a DNA sequence so that each pair of bits is replaced by a DNA code. For example, (00) is replaced by (A), (01) by (C), (10) by (G), and (11) by (T), giving the four sequences ($k_1$, $k_2$, $k_3$, and $k_4$). The addition operation is applied to the first and second sequences ($k_1$, $k_2$), while the subtraction operation is applied to the third and fourth sequences ($k_3$, $k_4$). Finally, the XOR process is performed on the results of the previous subtraction and addition stages. The results of these operations are used for generating a set of S-boxes by getting each digit as one cell in the S-box, and this cell should not be repeated before. The process continues until unique 256 values are placed in the S-box. Table 1 denotes the results of applying the suggested method for creating S-box.

Figure 4. Block diagram of encryption algorithm

3.1.4. Substitution
The S-box achieves the principle of substitution. In this stage, the permuted pixels in the previous step will be substituted based on new S-boxes have been generated, where three permuted vectors merge into one vector and are split into blocks, each block having a size of 8x8 bytes. Each block is converted into a hexadecimal string and substituted by one S-box. Based on a ($w_i$) chaotic sequence for the selection No. of S-box, ($w_i$ mod $n$), where $w_i$ represents the No. of S-box. After the S-box selection, each two-digit hexadecimal is substituted in the S-box. To change the value of each pixel, the first hexadecimal digit is given to the rows and the second hexadecimal digit is given to the columns. The value resulting from the intersection of the row and column in the s-box is put in the new matrix. The previous steps are repeated until all the blocks are completed and the get the image confusion. Algorithm 1 shows the total steps of the encryption method.
The chaotic algorithm is used to select the inverse S-boxes. The encryption phase is used to select the inverse S-boxes in the decryption stage, and $k_2$, $k_3$ are generated by iterating the proposed chaotic system with parameters and initial conditions to

Create four series $x_i$, $y_i$, $z_i$, and $w_i$, size of sequences $\geq S$

Step 4: split color image (im) into R, G, B bands

Step 5: Each color channel (Red, Green, and Blue) are reshape to ($V_R$, $V_G$, and $V_B$) vector

Step 6: Sort ($x_i$, $y_i$, $z_i$) sequences in ascending order, get the index, and find the position of each sequence in the index ($k_1$, $k_2$, $k_3$)

Step 7: Using $k_1$ to permute the vector ($V_R$) // (Red scrambling) using $k_2$ to permute the vector ($V_G$) // (Green scrambling) using $k_3$ to permute the vector ($V_B$) // (Blue scrambling)

Step 8: Merging the three ($V_R$, $V_G$, and $V_B$) vectors to one ($V_{RGB}$) vector

Step 9: divide the ($V_{RGB}$) vector to blocks of size 8x8

Step 10: using the 4th sequence $w_i$ as index ($w_i$ mod n) of dynamic S-Boxes, to select the No. of S-box // n the number of S-boxes produced

Step 11: After selecting the No. of S-box, for each block we substitute each value that denotes the address of the row and column in the S-box and get the new value by the intersection of the row and column.

Step 12: Continue until all the blocks are complete to get the confused image (Vsbox).

Step 13: split Vsbox into three vectors VsR, VsG, and VsB

Step 14: reshape VsR, VsG and VsB into 2D ($m_R$, $m_G$, and $m_B$) matrix

Step 15: combine ($m_R$, $m_G$, and $m_B$) to create 3-D of matrix represented the enciphered image

End

3.2. Decryption stage

Decryption is the inverse operation of encryption. The inputs to this stage are the encrypted image, initial condition values, and parameters, while the output is the extracted image. At first, the chaotic sequences are generated by iterating the proposed new chaotic system to create sequences: $x_i$, $y_i$, $z_i$, and $w_i$. These sequences are based on the same values of parameters and primary values as the encryption phase. Second, in the inverse substitution, the encrypted image is split into three bands and reshaped into vectors. The vectors are divided into blocks of size 8x8. Then each block is substituted by one from an inverse S-box. The chaotic $w_i$ sequence is used to select the inverse S-box ($w_i$ mod n), which is the number (NO. of inverse S-boxes). Thirdly, the inverse permutations are performed by the index of the chaotic sequences ($k_1$, $k_2$, and $k_3$) and the image vectors ($V_{IR}$, $V_{IB}$, and $V_{IB}$) to get the re-permutated image. Lastly, the image's vectors are combined and turned into a matrix that shows the extracted image. Figure 5 shows the total steps in the decryption stage, and algorithm 2 displays the main steps of decryption:

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Read color image (PI)</td>
</tr>
<tr>
<td>2</td>
<td>M&lt;height of PI N&lt;width of PI S&lt; MxN</td>
</tr>
<tr>
<td>3</td>
<td>Iterate proposed chaotic system with parameter and initial conditions to</td>
</tr>
<tr>
<td>4</td>
<td>Create four series $x_i$, $y_i$, $z_i$, and $w_i$, size of sequences $\geq S$</td>
</tr>
<tr>
<td>5</td>
<td>split color image (im) into R, G, B bands</td>
</tr>
<tr>
<td>6</td>
<td>Each color channel (Red, Green, and Blue) are reshape to ($V_R$, $V_G$, and $V_B$) vector</td>
</tr>
<tr>
<td>7</td>
<td>Sort ($x_i$, $y_i$, $z_i$) sequences in ascending order, get the index, and find the position of each sequence in the index ($k_1$, $k_2$, $k_3$)</td>
</tr>
<tr>
<td>8</td>
<td>Using $k_1$ to permute the vector ($V_R$) // (Red scrambling) using $k_2$ to permute the vector ($V_G$) // (Green scrambling) using $k_3$ to permute the vector ($V_B$) // (Blue scrambling)</td>
</tr>
<tr>
<td>9</td>
<td>Merging the three ($V_R$, $V_G$, and $V_B$) vectors to one ($V_{RGB}$) vector</td>
</tr>
<tr>
<td>10</td>
<td>divide the ($V_{RGB}$) vector to blocks of size 8x8</td>
</tr>
<tr>
<td>11</td>
<td>using the 4th sequence $w_i$ as index ($w_i$ mod n) of dynamic S-Boxes, to select the No. of S-box // n the number of S-boxes produced</td>
</tr>
<tr>
<td>12</td>
<td>After selecting the No. of S-box, for each block we substitute each value that denotes the address of the row and column in the S-box and get the new value by the intersection of the row and column.</td>
</tr>
<tr>
<td>13</td>
<td>Continue until all the blocks are complete to get the confused image (Vsbox).</td>
</tr>
<tr>
<td>14</td>
<td>split Vsbox into three vectors VsR, VsG, and VsB</td>
</tr>
<tr>
<td>15</td>
<td>reshape VsR, VsG and VsB into 2D ($m_R$, $m_G$, and $m_B$) matrix</td>
</tr>
<tr>
<td>16</td>
<td>combine ($m_R$, $m_G$, and $m_B$) to create 3-D of matrix represented the enciphered image</td>
</tr>
</tbody>
</table>

![A new four-dimensional hyper-chaotic system for image encryption (Huda R. Shakir)](image-url)
Figure 5. Block diagram of decryption algorithm

Algorithm 2. Decryption
Input: Encrypted image (imx), initial values: Y0, X0, Z0, W0, parameters: b, d, a, c, g, e, h, f, j, i
Output: Original image (Ex) of size (h×w×3)

Begin
Step 1: read encrypted image (imx)
Step 2: m ← height of imx
     n ← width of imx
     S ← m×n
Step 3: Iterate proposed chaotic system with parameter and initial conditions to
       Create four series x_i, y_i, z_i, and w_i, size of sequences ≥ S
Step 4: split image (imx) into three R, G, B bands
Step 5: Each color channel (Red, Green, and Blue) are reshape to (IV_R, IV_G, and IV_B) vector
Step 6: Merge IV_R, IV_G, and IV_B to one (IV_RGB) vector
Step 7: divide the (IV_RGB) vector to blocks of size 8×8
Step 8: using the 4th sequence w_i as index (w_i mod n) of inverse dynamic S-Boxes, to
       select the No. of inverse S-box // n the number of inverse S-boxes produced
Step 9: After selecting the No. of inverse S-box, for each block we substitute each
       value that denotes the address of the row and column in the S-box and get
       the new value by the intersection of the row and column.
Step 10: Continue until all the blocks are complete to get the Re-confused image(IVsbox)
Step 11: split (IVsbox) into three vectors VR, VG, and VB
Step 12: Sort (xi, yi, zi) sequences in ascending order, get the index, and find the
       position of each sequence in the index (k1, k2, k3).
Step 13: Using (k1) to re-permutate the vector(VR)
        using (k2) to re-permutate the vector(VG)
        using (k3) to re-permutate the vector(VB)
Step 14: reshape VR, VG and VB into 2D matrices MR, MG, and MB
Step 15: combine MR, MG, and MB to create 3D matrix represented the ciphered image(Ex)
End
4. RESULTS AND ANALYSIS

In order to illustrate our algorithm’s security and efficiency, some security analysis has been implemented on the suggested enciphering algorithm. In this paper, several standard images (256×256×3 and 512×512×3 pixels) have been tested. Furthermore, the proposed algorithm is simulated in MATLAB R2020a, and our testing results were compared with those of other encryption methods of the same type. These analyses are discussed.

4.1. Key space analysis

A cryptosystem’s key space should be at least $2^{100}$ to withstand brute force attacks [20], [21]. If the precision is in the range of $10^{-14}$, then the key space is $10^{196} \approx 2^{627}$. Therefore, the key space is large enough to withstand brute force attack. The key space comparative results between the proposed method and the related works are shown in Table 2.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Key space</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed method</td>
<td>$2^{627}$</td>
</tr>
<tr>
<td>Ref. [15]</td>
<td>$2^{299}$</td>
</tr>
<tr>
<td>Ref. [16]</td>
<td>$2^{425}$</td>
</tr>
<tr>
<td>Ref. [17]</td>
<td>$\approx 4 \times 10^{37}$</td>
</tr>
</tbody>
</table>

4.2. Key sensitivity analysis

A secure encryption system should be extremely sensitive to small changes in the key. In other words, the encryption technique should be able to withstand brute-force attacks and must be sensitive to any changes in the keys [22]. As explained in Figure 6, the key sensitivity for images is Figure 6(a) show the original image, Figure 6(b) show an encrypted image, Figure 6(c) show a decrypted image with the correct key, and Figure 6(d) show a decrypted image with the incorrect key.

![Figure 6](image)

Figure 6. The key sensitivity for images (a) original image, (b) encrypted image, (c) decrypted image with the correct key, and (d) decrypted image with the incorrect key

4.3. Histogram analysis

To stop the attacker from extracting any data contained in the plain image, any statistical relationship or similarity between the plain and decrypted images should be avoided. From a histogram analysis of an image, the statistical properties of that image can be derived. The plain and ciphered images must have completely different statistical properties [23]. Figure 7 displays the histogram of images tested before and after encryption.

4.4. Entropy analysis

Entropy is the significant characteristic that reflects information’s randomness and unpredictability [24], [25]. The entropy of the cipher image should ideally be 8 [11]. The entropy of $H(s)$ can be calculated:

$$H(s) = - \sum_{i=0}^{N-1} p(s_i) \log_2 p(s_i)$$

(2)

where $N$ denotes the number needed to represent the symbol $s_i$, $s$ denotes the source, and $P(s_i)$ is the symbol’s probability. Table 3 shows the entropy values of encryption images.
Figure 7. The histogram of image tested before and after encryption
A new four-dimensional hyper-chaotic system for image encryption

Huda R. Shakir

Table 3. The entropy test of images

<table>
<thead>
<tr>
<th>Name image</th>
<th>Entropy of original images</th>
<th>Entropy of encryption</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lina</td>
<td>7.7516</td>
<td>7.99857</td>
</tr>
<tr>
<td>palace</td>
<td>7.09016</td>
<td>7.99896</td>
</tr>
<tr>
<td>Yacht</td>
<td>7.6035</td>
<td>7.99802</td>
</tr>
<tr>
<td>sailboat on lake</td>
<td>7.7632</td>
<td>7.99897</td>
</tr>
</tbody>
</table>

Table 4. Correlation coefficients analysis

<table>
<thead>
<tr>
<th>Direction</th>
<th>Lena</th>
<th>palace</th>
<th>Yacht</th>
<th>Sailboat</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizontal</td>
<td>original</td>
<td>0.96438</td>
<td>0.89285</td>
<td>0.97313</td>
<td>original</td>
</tr>
<tr>
<td></td>
<td>encryption</td>
<td>-0.00020</td>
<td>0.00036</td>
<td>0.00068</td>
<td>encryption</td>
</tr>
<tr>
<td>vertical</td>
<td>original</td>
<td>0.98031</td>
<td>0.92422</td>
<td>0.96652</td>
<td>original</td>
</tr>
<tr>
<td></td>
<td>encryption</td>
<td>-0.00045</td>
<td>0.00130</td>
<td>0.00356</td>
<td>encryption</td>
</tr>
<tr>
<td>diagonal</td>
<td>original</td>
<td>0.94832</td>
<td>0.84829</td>
<td>0.93995</td>
<td>original</td>
</tr>
<tr>
<td></td>
<td>encryption</td>
<td>-0.00474</td>
<td>-0.00612</td>
<td>-0.00141</td>
<td>encryption</td>
</tr>
</tbody>
</table>

4.5. Correlation coefficients analysis

Correlation coefficient analysis is used to determine the degree of similarity between plaintext and ciphertext images [1], [26]. Table 4 shows the correlation coefficients analysis. The test was applied to red, green, and blue colors, respectively, between an original image and an encrypted image. As shown in Figure 8, the correlation is applied in horizontal, which tests the pixel with a neighbor in a row, vertical, which tests the pixel with a neighbor in a column, and diagonal, which tests the pixel with a neighbor pixel in the diagonal.

![Correlation coefficients of plain and encrypted image](image)

4.6. Number of pixels change rate (NPCR) and unified average changing intensity (UACI) analysis

Two differential assault metrics are used to evaluate how vulnerable the original data is to slight alterations: NPCR as well as UACI. Suppose the enciphered image is (C and C’) before and after changing one pixel in the original image [27]–[29]. The results of applying the proposed method on tested images are represented in Table 5. The (3) and (4) [30], express the NPCR and UACI formulas, respectively:

\[
NPCR = \frac{\sum_{i,j} D(i,j)}{W \times H} \times 100\% \tag{3}
\]

\[
UACI = \frac{1}{W \times H} \left[ \sum_{i,j} \left| \frac{C(i,j) - C'(i,j)}{255} \right| \right] \times 100\% \tag{4}
\]
4.7. Peak signal-to-noise ratio (PSNR) and mean square error (MSE) analysis

The PSNR and MSE are two of the more commonly used tests for image encryption techniques; the PSNR may be used to evaluate an encrypting method. It is a metric that indicates the difference in pixel values between the plain and cipher images. A lower PSNR value indicates a higher encoding quality. The PSNR and MSE formulas are [31]:

$$\text{PSNR} = 20 \log_{10} \left( \frac{255}{\sqrt{\text{MSE}}} \right) \text{dB}$$  \hspace{1cm} (5)

$$\text{MSE} = \frac{1}{MN} \sum_{i,j} (p_0(i,j) - p_1(i,j))^2$$  \hspace{1cm} (6)

where $M$ and $N$ represent the height and width of images. For the original and ciphered images, the intensity values of the pixels are $P_0(i,j)$ and $P_1(i,j)$. According to the results, a high MSE value and a low PSNR value between the original and encrypted images indicate desirable encryption quality, as shown in Table 6.

Table 5. UCAI and NPCR result

<table>
<thead>
<tr>
<th>Image</th>
<th>NPCR</th>
<th>UCAI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>99.6367</td>
<td>33.0305</td>
</tr>
<tr>
<td>Palace</td>
<td>99.6329</td>
<td>29.4177</td>
</tr>
<tr>
<td>Yaght</td>
<td>99.6183</td>
<td>31.0267</td>
</tr>
<tr>
<td>Sailboat</td>
<td>99.6152</td>
<td>34.0265</td>
</tr>
</tbody>
</table>

Table 6. The MSE and PSNR test

<table>
<thead>
<tr>
<th>Image</th>
<th>MSE</th>
<th>PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>8914.693</td>
<td>3.8508</td>
</tr>
<tr>
<td>Palace</td>
<td>8327.430</td>
<td>3.4910</td>
</tr>
<tr>
<td>Yaght</td>
<td>9262.862</td>
<td>3.2850</td>
</tr>
<tr>
<td>Sailboat</td>
<td>10171.02</td>
<td>2.9726</td>
</tr>
</tbody>
</table>

4.8. Comparison results

The proposed method is compared to other algorithms in order to see if it satisfies general requirements and how well it performs. Lena is used as the test image. Entropy, key space, NPCR, UACI, and correlation coefficient are utilized for the comparison. By looking at the algorithms, the suggested method is more secure against differential, statistical, and brute-force attacks than the methods that are currently being used. The results of the comparison are listed in Table 7.

Table 7. Comparison results of the proposed method with other methods

<table>
<thead>
<tr>
<th>Image</th>
<th>method</th>
<th>Key space</th>
<th>Entropy</th>
<th>NPCR</th>
<th>UACI</th>
<th>Correlation.H</th>
<th>Correlation.V</th>
<th>Correlation.D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encrypted Lena</td>
<td>proposed</td>
<td>$2^{67}$</td>
<td>7.9985</td>
<td>99.6367</td>
<td>33.0305</td>
<td>-0.00020</td>
<td>-0.00045</td>
<td>-0.00474</td>
</tr>
<tr>
<td>Ref. [15]</td>
<td>$2^{59}$</td>
<td>7.9984</td>
<td>99.6094</td>
<td>33.4635</td>
<td>-0.00216</td>
<td>0.00103</td>
<td>0.0004</td>
<td></td>
</tr>
<tr>
<td>Ref. [16]</td>
<td>$2^{55}$</td>
<td>7.9912</td>
<td>99.6235</td>
<td>33.4620</td>
<td>0.000617</td>
<td>-0.00053</td>
<td>-0.00041</td>
<td></td>
</tr>
<tr>
<td>Ref. [17]</td>
<td>$4 \times 10^{12}$</td>
<td>7.9974</td>
<td>99.62</td>
<td>31.03</td>
<td>-0.0026</td>
<td>0.0031</td>
<td>-0.0043</td>
<td></td>
</tr>
<tr>
<td>Ref. [20]</td>
<td>$2^{60}$</td>
<td>7.929</td>
<td>99.665</td>
<td>32.4966</td>
<td>-0.0921</td>
<td>-0.0372</td>
<td>-0.1013</td>
<td></td>
</tr>
<tr>
<td>Ref. [28]</td>
<td>-</td>
<td>7.9976</td>
<td>99.64</td>
<td>28.66</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

4.9. Time consuming

The execution time is also an important factor in evaluating the performance of an encryption algorithm. The encryption algorithm is more efficient when the execution time is less. Table 8 shows the average consumption time for two-stage encryption and decryption.

Table 8. Time-consuming for encryption/decryption technique

<table>
<thead>
<tr>
<th>Image size</th>
<th>Encryption Time (sec)</th>
<th>Decryption time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>256x256</td>
<td>0.58233</td>
<td>0.79361</td>
</tr>
<tr>
<td>512x512</td>
<td>1.29370</td>
<td>1.36520</td>
</tr>
</tbody>
</table>

5. CONCLUSION

In this study, a robust color image encryption algorithm built on a novel chaotic four-dimensional system has been suggested to achieve a high level of security, fast speed, and high performance. A variety of security analyses, including entropy analysis and histogram analysis, have been performed on the proposed scheme. The results of the simulations and performance measurements have shown that the encryption effect on security and reliability of our suggested method is good and it is well suited for image encryption. The proposed method has a large key space of $(2^{67})$ and a high sensitivity to a slight change in the secret key of decrypted images, a fairly uniform histogram, and entropy values close to the ideal value of 8, mean values.
for NPCR and UACI values of (99.63) and (33.03), respectively, and a very short time for encryption and decryption. In future work, the proposed method could be applied to other types of data, such as video and audio data, also by merging the proposed system with other techniques like watermarking or steganography techniques.

REFERENCES


BIOGRAPHIES OF AUTHORS

Huda R. Shakir received a B.Sc. in Computer Sciences from Mustansiriya University, Baghdad, Iraq. Currently, she is studying M.Sc. in Computer Science, College of Education, Department of Computer Science, Mustansiriya University, as well as working in the Iraqi Ministry of Education. She can be contacted at hudarashid@uomustansiriyah.edu.iq.

Sadiq A. Mehdi received the B.Sc. in Mathematical Sciences from Mustansiriya University, Baghdad, Iraq in 1996, M.Sc. in Applied Mathematics Sciences-Modeling and Simulation from Al al-Bayt University, Jordan in 2002, and Ph.D. in Applied Mathematics/Data Cryptography from University of Mustansiriya, Baghdad, Iraq in 2011. Current position and Functions: computer science from Mustansiriya University. His research interest is in the fields of Dynamical system, Chaotic system, Chaotic Encryption and Modeling and Simulation. He can be contacted at sadiqmehdi71@uomustansiriyah.edu.iq.

Anwar A. Hattab received the B.Sc in Computer Science from Baghdad University and her MSc degree in Network Management in 2003 from the Iraq Commission for Computer and Informatics, Institute for Post Graduate Studies in Informatics. Currently she is a lecturer in computer science. Anwar has more than 18 years of experience and has supervised MSc and BSc final year project. Her research interests include cryptography, image processing, data security, network security, and databases. She can be contacted at email: anwarahbas76@uomustansiriyah.edu.iq.