Optimized Color Image Encryption Using Arnold Transform, URUK Chaotic Map and GWO Algorithm

Qutaiba K. Abed¹, Waleed A. Mahmoud Al-Jawher²

¹ Informatics Institute for Postgraduate Studies, Iraqi Commission for Computers and Informatics, Baghdad, Iraq.
² College of Engineering, Uruk University, Baghdad, Iraq

phd202130682@iips.edu.iq

Abstract A new image encryption algorithm based on the Arnold transform and URUK chaotic maps is proposed to deal with the issues of inadequate security and low encryption efficiency. Colored images consist of three linked channels used in the scheme. This method uses different keys to break the correlations between adjacent pixels in each channel. First, the plain image is split into RGB channels to encrypt each channel separately. Second, the Arnold transform performs pixel permutation, resulting in scrambled channels. third, the URUK chaotic maps generate three key vectors to perform pixel diffusion, resulting in diffused channels used as input for the following step. Finally, the GWO shuffles each channel independently, to get the minimum correlation between image pixels, which are then merged to obtain a cipher image. This method generates the cipher image with great unpredictability and security. The security is evaluated using various measures. The results demonstrated a high level of security attained by successfully encrypting colored images. Recent encryption algorithms are compared in terms of entropy, correlation coefficients, and attack robustness. The proposed method provided outstanding security and outperformed existing image encryption algorithms.

Keywords: Arnold transform, GWO, Fnet, URUK chaotic map

1. INTRODUCTION

The advancement of multimedia technologies and the rise of the internet have led to a significant increase in information transmission, particularly digital images. These images play a vital role in various fields like remote sensing, medicine, and military communication. However, they can also contain sensitive personal or confidential information. Unauthorized access to this critical information can cause problems for both individuals and countries that own it [1,2]. This highlights the importance of protecting sensitive information. Encryption [3,4], steganography [5,6], data hiding [7,8], and watermarking are some of the common methods used to secure digital images [9,10].

While many encryption techniques exist, traditional methods designed for text aren't ideal for digital images, especially color ones. This is because color images have unique properties like redundancy, strong correlations between pixels, and large file sizes [11]. To ensure secure image encryption, reliable and robust methods are crucial. Researchers have proposed various approaches using different technologies, such as chaos theory [12, 13], substitution boxes [14], and even DNA [15]. This study specifically explores the use of chaos-based encryption for color images. Chaos theory is particularly useful for image encryption because even tiny changes in a chaotic system can lead to vastly different outcomes [16–18]. Additionally, chaotic systems are excellent at evenly covering a space (ergodicity), have a predictable past (determinism), and generate seemingly random numbers (pseudo-randomness) – all qualities ideal for encryption [19–22].

Image encryption aims to scramble an image using a secret key, making it appear random and unreadable. The same key is then used to decrypt the image and recover the original content. Following principles established by Shannon's information theory, effective encryption relies on two key techniques: permutation and diffusion [23]. Fridrich was a pioneer in chaos-based image encryption. His approach used chaotic maps to scramble the order of pixels in an image, relying on the iterative values generated by these maps [24]. As Shannon suggested, most image encryption methods involve both permutation and diffusion. Permutation rearranges the positions of pixels without changing their values, effectively reducing the correlation between them. Diffusion then alters the actual pixel values, creating the final encrypted image (cipher image). Combining permutation and diffusion provides a higher level of security [25–32].

The field of chaos-based image encryption has seen a steady rise in proposed techniques. For example, Quan et al. developed an encryption algorithm that utilizes chaotic maps with specific statistical properties [33]. Similarly, Wang et al. built upon Shannon and Fridrich's principles by designing a
new encryption method using complex chaotic systems from Lorenz and Chen [34]. Another contribution came from Ali et al., who introduced a novel hyper-chaotic map named 2D-HLCM. This map combines elements from logistic, Henon, iterative infinite folding, and infinite collapse maps, and it finds application in image encryption [35]. As mentioned earlier, encryption relies on manipulating both the order and the values of pixels. Sun et al. presented a unique chaotic system using special memristors for image encryption [36]. Ali and Ali proposed a three-step method for color image encryption [37]. First, they use a chaotic map to shuffle the pixel order in the original image (permutation). Next, they substitute pixel values using a substitution box derived from the same chaotic map. Finally, they perform an XOR operation to alter the pixel values themselves. Xiang and Liu improved upon the logistic map, using their enhanced version in a new color image encryption scheme [38]. Mondal and Mandal introduced a novel method that combines a pseudorandom number generator with genetic algorithms for image encryption [39]. Bouteghrine et al. created a new 3D chaotic system specifically designed for color image encryption [40]. Mou et al. took an innovative approach by combining image compression with encryption using a hyper-chaotic map. This not only enhances security but also reduces storage and transmission costs [41].

Despite their practicality, most chaos-based image encryption algorithms have both strengths and weaknesses. An ideal color image encryption algorithm should possess several key features: a vast and complex key space to thwart statistical attacks, the ability to handle images of any size, and a significant reduction in the original image's pixel correlations. To evaluate the effectiveness of these algorithms and identify areas for improvement, numerous studies have explored methods to break them (cryptanalysis) [42-48]. This ongoing research helps to strengthen the security of chaos-based image encryption schemes. Previous methods using a single 1D chaotic map for all color channels (red, green, and blue) had security weaknesses, especially when encrypting multiple images with the same key. This approach made it vulnerable to attacks that exploit correlations between adjacent pixels. This paper proposes a novel encryption technique for color images. It achieves this through a two-step process known as confusion and diffusion. The contribution lies in its combination of techniques:

- **URUK Chaotic Maps and Arnold Transform:** These work together to disrupt the original image's structure. The Arnold transforms shuffles pixels within each color channel, while URUK keys (X, Y, Z) significantly alter the values in these shuffled channels, creating entirely different encrypted versions.

- **Grey Wolf Optimization (GWO) Algorithm:** This further enhances security by shuffling pixels across all three channels (red, green, and blue). This minimizes any remaining correlations and ensures a highly random encrypted image.

By exploiting the chaotic nature of URUK maps, GWO, and Arnold transforms, the proposed method effectively breaks down correlations between neighboring pixels and across RGB channels. This significantly improves security by making the encrypted image resistant to common attacks like statistical and differential analysis. Additionally, the vast key space provided by the combined elements strengthens the encryption process.

The remainder of the paper as follows: Section 2 URUK chaotic system. Section 3 presents the Fnet transformer. Section 4 Gray Wolf Optimization (GWO). Section 5 explains the Encryption process, section 6 describes the decryption process, section 7 Discusses experiential results and analysis, and Section 8 shows some conclusions.

## 2. URUK Chaotic System

The Uruk chaotic system is a relatively new mathematical model that exhibits chaotic behavior. This means it’s a system that’s sensitive to initial conditions, and unpredictable in the long term. The system operates in four dimensions (often denoted as X, Y, Z, and W) and evolves in discrete steps rather than continuously. It exhibits intricate and unpredictable behavior over time, even with small changes in its starting conditions. Due to its complex and unpredictable nature, the Uruk system has potential applications in cryptography and image encryption. The unpredictable outputs can be used to scramble data, making it unreadable to unauthorized users.

\[
X_{(n+1)} = 1 - (X_n \times Y_n \times Z_n \times W_n) - X_n^2 - Y_n^2 - a \times \tan(Z_n) - W_n^2
\]

\[
Y_{(n+1)} = X_n - b \times \tan(Z_n)
\]

\[
Z_{(n+1)} = Y_n - c \times \tan(Z_n)
\]

\[
W_{(n+1)} = X_n - d \times W_n
\]

(1)

A mathematical system tracks four elements (x, y, z, and w) that can behave unpredictably. Certain values (a, b, c, and d) influence this erratic behavior. The system's equations are tweaked with trigonometric functions and complex interactions to make its outputs even more random [49].

## 3. Fnet Transformer

FNet streamlines the Transformer architecture by removing the self-attention mechanism in each encoder layer. Instead, it utilizes a Discrete Wavelet Transform (DWT) mixing sublayer [50-73]. This sublayer applies a 2D DWT to the data, achieving similar results without the computational burden of attention [74,75].
4. Gray Wolf Optimization (GWO)

GWO is a population-based optimization algorithm inspired by grey wolf social behavior during hunting.

- Social Hierarchy: It simulates a pack structure. Each candidate solution corresponds to a wolf. The "fittest" solution (one closest to the optimal value) is designated as the Alpha wolf. Beta and Delta wolves represent good solutions, while Omega represents the least fit.

- Hunting Phases: The optimization process mimics the hunting stages:
  1. Search: Wolves (solutions) update their positions based on the locations of Alpha, Beta, and Delta. This guides the search towards promising areas.
  2. Encircling Prey: Wolves strategically encircle the perceived prey (optimal solution) based on the positions of Alpha, Beta, and Delta. This guides the search towards promising areas.
  3. Attacking Prey: Wolves converge towards the prey to exploit it (find the optimal solution). This convergence process is mathematically modeled to iteratively improve solutions.

- Mathematical Modeling: The positions of wolves are represented by vectors, and their movements are controlled by mathematical equations that consider the social hierarchy and hunting behavior.

GWO offers a powerful tool for solving complex optimization problems in various fields like engineering design, machine learning, and scheduling [76-82].

5. Encryption Process

Figure 2 illustrates the primary structural diagram of our proposed algorithm. The proposed image encryption method consists of two stages: Confusion, where pixel positions are scrambled depending on the Arnold transform and gray wolf optimization algorithm. The second requires diffusion over the pixels of the RGB channels using Fnet and URUK chaotic map. The detailed encryption steps are described below:

1. Input color image size of (256 × 256) pixels.
2. separate the image (RGB format) into its individual red, green, and blue channels. Each of these channels with a size of 256 pixels by 256 pixels.
3. Applying Arnold transform to each channel separately to get scrambled channels.
4. Generate the initial keys for the URUK chaotic map as follows
   a. Convert the color image into a grayscale image.
   b. The image is fed into a hashing function called SHA512. This function scrambles the image data into a unique 512-bit string.
   c. The 512-bit hash is divided into 64 groups of 8 bits each. Each group is essentially a number between 0 and 255 (represented in decimal).
   d. Four key values, X, Y, Z, and W, are calculated using the following mathematical equations that likely involve these 64 decimal numbers.

\[
key_1 = \sum_{i=1}^{16} H_i, \quad X = \text{mod}([key_1 \times 2^6.99])
\]
\[
key_2 = \sum_{i=17}^{32} H_i, \quad Y = \text{mod}([key_2 \times 2^6.99])
\]
\[
key_3 = \sum_{i=33}^{48} H_i, \quad Z = \text{mod}([key_3 \times 2^6.99])
\]
\[
key_4 = \sum_{i=49}^{64} H_i, \quad W = \text{mod}([key_4 \times 2^6.99])
\]
5. Applying URUK chaotic map to generate X, Y, Z vectors and apply Fnet to each vector as follows:
   a. Generate position encoding to each vector and add it to the original vector to get F vector.
   b. Apply the DWT to the vector.
   c. Apply a layer normalization to each vector to get F1 vector.
   d. Add normalized vector F1 to the F vector to get F2.
   e. Apply multi-layer perceptron (MLP) for F2 vector to get F3.
   f. Apply a layer normalization to F3 vector to get F4 vector.
   g. Add normalized vector F4 to the F3 vector to get F5 vector.

\[\text{Figure 1: FNet architecture with N encoder blocks.}\]
h. apply the final multi-layer perceptron (MLP) for F5 vector to F6 vector.  
i. Apply the following equation to F6 vector to get the final quantified vector between [0-255] which use in the diffusion process.  
\[ O = \text{Round} \left( \text{mode} \left( Y \times 10^9, 256 \right) \right) \]

6. Applying diffusion process to each channel using X, Y, Z vectors  
\[ \text{Im} = \text{Xor} \left( \text{[red, green, blue], [X, Y, Z]} \right) \]  

7. Applying the GWO to shuffle the position of each channel as follows  
a. convert the channel to 1D dimension  
b. generate the population of wolves  
c. sort the position of each wolf and get the index  
d. shuffle the position of each channel based on the indexes of wolves depending on the following objective function  
\[ \text{Min FC} = \text{Correlation} \left( \text{channel} \right) \]  
continue for all iteration to get the minimum correlation between pixels to get the final cipher image

6. The Decryption Process

The decryption process is the same as the encryption process, but in reverse. The image is divided into three channels (red, green and blue) where the shuffle of GWO is reversed for each channel. Next the diffusion process is reversed. Finally, the scrambling is reversed by Arnold transform for each channel then colors are mixed to obtain the plain image.

7. The Experiential Results And Analysis

Security analysis is a crucial step to evaluate the performance of an encryption algorithm. It assesses how well the algorithm resists various attacks that aim to retrieve the original data from the encrypted ciphertext. Here is the common breakdown of security analysis for image encryption algorithms:

7.1 Keyspace analysis

Brute-force attacks are a major concern for encryption algorithms. To address this, we have designed our algorithm with a large key space, exceeding \(2^{100}\) (the recommended minimum size according to [46]). This vast number of possible keys, as shown in Table 1, makes our algorithm highly resistant to brute-force attempts to crack the encryption.
7.2 Information entropy

Information entropy is an important indicator to describe the uncertainty of image information, which quantifies the distribution of the image’s grayscale values [17]. Generally speaking, the higher the information entropy value, the higher the degree of disorder in the image. The formula of information entropy is as follows.

\[
H(s) = -\sum_{i=1}^{L} p(x_i) \log_2 p(x_i),
\]

where \( L \) is the grayscale grade of the image, and \( p(x_i) \) is the probability of the grayscale value \( x_i \).

For 8-bit noise type grayscale images, the ideal value of information entropy is 8. The information entropy of different plain images and their corresponding cipher images are listed in Table 2. As can be seen, values of the information entropy of all encrypted images are close to 8. Table 3 lists the comparison results with other algorithms for Lena. It is obvious that the proposed algorithm owns a larger information entropy compared with other algorithms, which means the cipher images encrypted by the proposed algorithm have a stronger randomness. Thus, the proposed algorithm can resist statistical attacks based on entropy.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Key spaces</td>
<td>(2^{256})</td>
<td>(2^{99})</td>
<td>(2^{213})</td>
<td>(2^{186})</td>
</tr>
</tbody>
</table>

Table 1. Keyspace for different algorithms.

Table 2: Information entropy of the proposed method.

<table>
<thead>
<tr>
<th>Image</th>
<th>Original</th>
<th>Cipher</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R</td>
<td>G</td>
</tr>
<tr>
<td>Lena</td>
<td>7.3183</td>
<td>7.6042</td>
</tr>
<tr>
<td>Baboon</td>
<td>7.6058</td>
<td>7.3581</td>
</tr>
<tr>
<td>Pepper</td>
<td>7.3009</td>
<td>7.5570</td>
</tr>
<tr>
<td>Tree</td>
<td>7.2587</td>
<td>7.6143</td>
</tr>
</tbody>
</table>

Table 3: Comparison of information entropy.

<table>
<thead>
<tr>
<th>Method</th>
<th>R</th>
<th>G</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed</td>
<td>7.9969</td>
<td>7.9968</td>
<td>7.9972</td>
</tr>
<tr>
<td>Ref [86]</td>
<td>7.9973</td>
<td>7.9972</td>
<td>7.9966</td>
</tr>
<tr>
<td>Ref [87]</td>
<td>7.9974</td>
<td>7.9971</td>
<td>7.9973</td>
</tr>
<tr>
<td>Ref [88]</td>
<td>7.9972</td>
<td>7.9965</td>
<td>7.9962</td>
</tr>
</tbody>
</table>
Qutaiba K. Abed, Waleed A. Mahmoud Al-Jawher. 2024, Optimized Color Image Encryption Using Arnold Transform, URUK Chaotic Map and GWO Algorithm. *Journal port Science Research*, 7(3), pp. 219-236. [https://doi.org/10.36371/port.2024.3.3](https://doi.org/10.36371/port.2024.3.3)
7.3. Histogram Analysis

The distribution of image pixel values can be reflected by the image histogram. If the histogram of a cipher image is flat, information of the plain image is excellently hidden. Figure 3 shows the histograms of the images before and after encryption. It can be seen that the histograms of encrypted images become relatively flat. Therefore, the proposed algorithm can effectively resist statistical attacks.

7.4 Correlation Analysis of Adjacent Pixels

The plain image with effective information has a strong correlation between adjacent pixels. The ideal encryption algorithm can eliminate the correlation of adjacent pixels to resist statistical attacks. To ensure the reliability of the experiment of pixels are test the correlation in horizontal, vertical, and diagonal directions. As shown from Figure 4, the adjacent pixel distribution of the plain image is relatively concentrated, whereas the adjacent pixel distribution of the cipher image is noise-like. This means that the correlation of the plain image is greatly reduced. To quantitatively describe the correlation, the correlation coefficient is calculated as follows.

\[ r_{ij} = \frac{C_o(i) - C_o(j)}{\sqrt{D(i)D(j)}} \]  

The calculated correlation coefficients are shown in Table 4. It can be seen that the correlation coefficients of the cipher images have been greatly reduced, close to 0. The results compared with other algorithms as shown in Table 5. As can be seen, the correlation coefficients of Lena for the proposed algorithm are smaller in all three directions compared with other methods. The proposed algorithm can effectively remove the correlation of adjacent pixels, so it provides a high level of security to resist statistical attacks.

<table>
<thead>
<tr>
<th>Image</th>
<th>direction</th>
<th>R</th>
<th>G</th>
<th>B</th>
<th>R</th>
<th>G</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>H</td>
<td>0.9399</td>
<td>0.9417</td>
<td>0.8886</td>
<td>0.0005</td>
<td>0.0013</td>
<td>-0.0005</td>
</tr>
<tr>
<td></td>
<td>V</td>
<td>0.9682</td>
<td>0.9697</td>
<td>0.9385</td>
<td>-0.0003</td>
<td>0.0004</td>
<td>-0.0001</td>
</tr>
<tr>
<td></td>
<td>D</td>
<td>0.9086</td>
<td>0.9126</td>
<td>0.8352</td>
<td>0.0004</td>
<td>0.0001</td>
<td>0.0000</td>
</tr>
<tr>
<td>Baboon</td>
<td>H</td>
<td>0.9474</td>
<td>0.8728</td>
<td>0.9216</td>
<td>0.0007</td>
<td>-0.0002</td>
<td>-0.0009</td>
</tr>
<tr>
<td></td>
<td>V</td>
<td>0.9208</td>
<td>0.8380</td>
<td>0.9139</td>
<td>-0.0010</td>
<td>-0.0018</td>
<td>-0.0003</td>
</tr>
</tbody>
</table>
### Table 5: Comparison of the Correlation of Ciphertext Lena.

<table>
<thead>
<tr>
<th>Method</th>
<th>R</th>
<th>G</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed</td>
<td>H</td>
<td>0.0005</td>
<td>0.0013</td>
</tr>
<tr>
<td></td>
<td>V</td>
<td>-0.0003</td>
<td>0.0004</td>
</tr>
<tr>
<td></td>
<td>D</td>
<td>0.0004</td>
<td>0.0001</td>
</tr>
<tr>
<td>Ref [86]</td>
<td>H</td>
<td>0.0007</td>
<td>-0.0035</td>
</tr>
<tr>
<td></td>
<td>V</td>
<td>-0.0004</td>
<td>0.0023</td>
</tr>
<tr>
<td></td>
<td>D</td>
<td>0.0039</td>
<td>-0.0079</td>
</tr>
<tr>
<td>Ref [87]</td>
<td>H</td>
<td>-0.0154</td>
<td>-0.0096</td>
</tr>
<tr>
<td></td>
<td>V</td>
<td>-0.0102</td>
<td>0.0027</td>
</tr>
<tr>
<td></td>
<td>D</td>
<td>0.0159</td>
<td>-0.0162</td>
</tr>
<tr>
<td>Ref [88]</td>
<td>H</td>
<td>0.0073</td>
<td>-0.00054</td>
</tr>
<tr>
<td></td>
<td>V</td>
<td>-0.00508</td>
<td>0.00331</td>
</tr>
<tr>
<td></td>
<td>D</td>
<td>0.00311</td>
<td>0.00076</td>
</tr>
</tbody>
</table>
Figure 4: The correlation of colored-Lena plain-image and corresponding ciphered-image
7.5 Chosen/Known-Plaintext Attack Analysis

Chosen-plaintext and known-plaintext attacks are prevalent and high-threat types of attacks. The literature [59] indicated that an encryption algorithm with the capability to resist chosen-plaintext attacks can also resist known-plaintext attacks. Therefore, we only consider resisting chosen-plaintext attacks. Entropy In the proposed algorithm, we exploit the SHA-512 hash values of the plain image to generate the system parameters and initial values of the chaotic system, making the proposed algorithm highly sensitive to the plain image. Thus, when attackers use the proposed algorithm to encrypt slightly changed plain images, the encryption result obtained is totally different. Attackers cannot gain the desired information using special images. Furthermore, we perform bit-level exclusive-or operations between different bit-planes. Attackers are incapable of using special images to simplify the diffusion process.

7.6 Cropping Attack and Noise Attack Analysis

In the actual transmission process of the network, the images are at high risk of data loss or noise contamination. Therefore, a secure image encryption algorithm shall be robust against cropping attacks and noise. To test the anti-noise performance of the proposed algorithm, we add salt and pepper noise with different intensities to the cipher image, where the intensities are 0.01, 0.03, 0.05, and 0.1, respectively. The results are shown in Figure 6 a−h. It can be seen that the decrypted images contain some noises, but we can still recognize most of the information in the plain image by human eyes. The proposed algorithm is resistant to noise attacks. Thus, the proposed algorithm can effectively resist cropping attacks and noise attacks.

Figure 5. The results of cropping attack, the encrypted images (a, b, c, d) with data loss of (16×16) pixels, (32×32) pixels, (64×64) pixels, (128×128), respectively where the images (e, f, g and h) are the decrypted images with PSNR (34.1651, 28.4776, 22.5790 and 16.6564) respectively.
7.7 MSE and PSNR

A ciphered image should exhibit substantial deviation from its original form. Mean square error (MSE) quantifies the cumulative squared difference between the original and corresponding ciphered images. It can be calculated through:

$$MSE_{(P,E)} = \frac{1}{W \times H} \sum_{i=0}^{W} \sum_{j=0}^{H} (P(i,j) - E(i,j))^2$$  \hspace{1cm} (9)$$

where $P(i,j)$ is the value of the pixels of the plain image and $E(i,j)$ is the encrypted pixel value at position $(i,j)$ in the cipher image. The MSE value can serve as a criterion for assessing the encryption strength of a cryptosystem. The larger the MSE scale, the greater the encryption security. PSNR analysis is a way of deciding the encryption quality level; the higher the scale the closer the encrypted image is to the original image. Hence, a lower PSNR value indicates more robust encryption for a cryptosystem. It is calculated as follows:

$$PSNR = 20 \times \log_{10}[\frac{255}{\sqrt{MSE}}]$$  \hspace{1cm} (10)$$

The MSE and PSNR values in Table 6 between the plain image and the cipher image.

### Table 6: PSNR & MSE.

<table>
<thead>
<tr>
<th>Image</th>
<th>MSE R</th>
<th>MSE G</th>
<th>MSE B</th>
<th>PSNR R</th>
<th>PSNR G</th>
<th>PSNR B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>168.4700</td>
<td>88.3956</td>
<td>94.4988</td>
<td>7.8992</td>
<td>8.5331</td>
<td>9.5281</td>
</tr>
<tr>
<td>Baboon</td>
<td>126.3336</td>
<td>118.2624</td>
<td>102.7197</td>
<td>8.9241</td>
<td>9.4730</td>
<td>8.5792</td>
</tr>
<tr>
<td>Pepper</td>
<td>138.5282</td>
<td>106.0031</td>
<td>56.7380</td>
<td>9.1000</td>
<td>7.6646</td>
<td>7.6967</td>
</tr>
<tr>
<td>Aircraft</td>
<td>166.7014</td>
<td>166.9344</td>
<td>180.1188</td>
<td>8.1765</td>
<td>7.8837</td>
<td>7.9822</td>
</tr>
<tr>
<td>Tree</td>
<td>120.5613</td>
<td>114.3240</td>
<td>104.6394</td>
<td>9.5133</td>
<td>7.6103</td>
<td>7.5719</td>
</tr>
</tbody>
</table>
7.8 Differential attack

Two standard sensitivity measures are appointed to check the resisting level of the differential attack in this paper: unified average changing intensity (UACI) and the number of pixels change rate (NPCR). The definitions of UACI and NPCR are as follows:

\[
\text{UACI} = \frac{1}{w \times b} \times \sum_{x=0}^{w} \sum_{y=0}^{b} \frac{|C_1(x,y) - C_2(x,y)|}{255} \times 100\%
\]

\[
\text{NPCR} = \frac{1}{w \times b} \times \sum_{x=0}^{w} \sum_{y=0}^{b} D(x,y) \times 100\%
\]

Where

\[
D(x,y) = \begin{cases} 
0, & C_1(x,y) = C_2(x,y) \\
1, & C_1(x,y) \neq C_2(x,y)
\end{cases}
\]

The symbols \(C_1, C_2\) refer to the two cipher images whose corresponding plain image has one random pixel variance, and \((w,h)\) represents the number of rows and columns. In the 256 × 256 images, one pixel is selected randomly, and the pixel value is changed to value+1 and then encrypted. The calculated values of UACI and NPCR for different colored images are shown in Table 7. In addition, the values of UACI and NPCR of different algorithms are listed in Table 8.

Table 7: UACI and NPCR values for encrypted images.

<table>
<thead>
<tr>
<th>Image</th>
<th>UACI%</th>
<th>NPCR%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R</td>
<td>G</td>
</tr>
<tr>
<td>Lena</td>
<td>33.5010</td>
<td>33.5307</td>
</tr>
<tr>
<td>Baboon</td>
<td>33.3539</td>
<td>33.4655</td>
</tr>
<tr>
<td>Pepper</td>
<td>33.4465</td>
<td>33.4444</td>
</tr>
<tr>
<td>Aircraft</td>
<td>33.4727</td>
<td>33.4271</td>
</tr>
<tr>
<td>Tree</td>
<td>33.6059</td>
<td>33.4008</td>
</tr>
</tbody>
</table>

Table 8: Comparison of UACI and NPCR.

<table>
<thead>
<tr>
<th>Image</th>
<th>UACI%</th>
<th>NPCR%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R</td>
<td>G</td>
</tr>
<tr>
<td>Proposed</td>
<td>33.5010</td>
<td>33.5307</td>
</tr>
<tr>
<td>Ref [86]</td>
<td>33.5031</td>
<td>33.4968</td>
</tr>
<tr>
<td>Ref [87]</td>
<td>33.4128</td>
<td>33.4980</td>
</tr>
<tr>
<td>Ref [88]</td>
<td>33.0704</td>
<td>30.7620</td>
</tr>
</tbody>
</table>

8. CONCLUSION

In this paper, an image encryption technique is introduced. Its implementation involves the combination of URUK chaotic maps, Arnold transforms and GWO to introduce confusion and diffusion. Color separated into (red, green and blue) channel and a different key in each image is used for the confusion and diffusion processes. This process aided in breaking the correlation between neighboring pixels in each channel. The performance of the scheme was assessed using various measures. These comprised histogram, entropy, MSE, PSNR, correlation coefficient, key space and differential attack. The computed results indicate that the suggested technique is robust to all visual, statistical, differential, and brute-force attacks. Furthermore, the proposed image encryption technique provides comparable or greater security performance.
REFERENCES


[14] Ramakrishnan, B., et al.: Image encryption based on S-box generation constructed by using a chaotic autonomous snap system with only one equilibrium point. Multimedia Tools Appl. 83(8), 23509–23532 (2024)


