Comparative Analysis of Color Image Encryption-Decryption Methods Based on Matrix Manipulation

Rashad J. Rasras¹; Mohammed Abuzalata²; Ziad Alqadi³; Jamil Al-Azze⁴; Qazem Jaber⁵

¹Computer Engineering Department, Al Balqa’á Applied University, Amman, 11134, Jordan
E-mail: Rashad.Rasras@bau.edu.jo
²Computer Engineering Department, Al Balqa’á Applied University, Amman, 11134, Jordan
E-mail: abuzalata@bau.edu.jo
³Computer Engineering Department, Al Balqa’á Applied University, Amman, 11134, Jordan
E-mail: Natalia_maw@yahoo.com
⁴Computer Engineering Department, Al Balqa’á Applied University, Amman, 11134, Jordan
E-mail: azzehjamil@gmail.com
⁵Computer Engineering Department, Al Balqa’á Applied University, Amman, 11134, Jordan
E-mail: qazemjaber@gmail.com

Abstract: Color image encryption-decryption is an important issue, because it is used in many important applications. This paper will introduce 3 methods of image encryption-decryption, these methods will be implemented and tested, and the obtained experimental results will be compared with experimental results of the proposed method in order to do some judgment regarding the efficiency and the security of the proposed method.

Keywords: Encryption time, decryption time, MSE, PSNR, private key, window.

1- Introduction

True color image is a 3D matrix, the first dimension represents the red color, the second one represents the green color, while the third one represents the blue color [1-41]. Digital color image is one of the most popular data type used in the internet as a result of data communication between the sender and receiver over the internet [4]. Color images are widely used by different users, and several applications need certain and consistent 'security in data communication and security in storing' [5], [6], such as medical imaging systems,
pay-TV, confidential video conferences, and military image communications, so the need for image encryption-decryption must have a priority with highest level.

Color encryption methods act as shown in figure (1) by selecting a private secret key and manipulating the original image in such a way to get the encrypted image.

The used method of encryption must destroy the original image in order to make it impossible to be understood by a third party or by unauthorized person, thus understanding level will be measured by the mean square error (MSE), or a peak signal to noise ratio (PSNR) between the original image and the encrypted one [7], [8], Where the value of the large error indicates the effectiveness of the method used and vice versa the small value of PSNR indicates the effectiveness of the method of encryption-decryption method, MSE between the original image and the decrypted one must lead to zero [7], [8] as shown in figure (2) and (3):

Figure (1): Encryption-decryption process

Figure (2): Destroying the encrypted image
Many methods of encryption-decryption are based on matrix multiplication [10, 11, 12, and 13], if we multiply the original image matrix by a secret key then we can get the encrypted image, and if multiply the encrypted image by the key inverse we can get the decrypted original matrix as shown in figures (4) and (5)[10], [11].
Colors from color images can be extracted, and each color may be encrypted, then the encrypted color image cab constructed from the 3 encrypted colors as shown in figure (6) [12], [13].

![Figure (6): Encrypting each color of the color image.](image)

**2- The Studied Methods**

Here we will describe 3 methods [14] which are used for color image encryption-decryption, and will a new one, this method will be tested and implemented and a comparative analysis between the four methods will be done.

**Method 1: Encryption individual color component:**

To do this we have to apply the following phases:

1) Phase 1: Private key generation
   This phase is to be implemented one time and it includes the following tasks:
   A. Generate a huge 2D matrix which is to be called key.
   B. Save the key.

2) Phase 2: Image encryption
   To do this we have to implement the following tasks:
   A. Get the original image.
   B. Extract the color components (Red, Green, and Blue).
   C. Load the key.
   D. Adjust the key to suit the 2D matrix.
   E. Get the encrypted Redenc, Greenenc, and Blueenc applying matrix multiplication of the key and each one of the 2D matrices.
   F. Compose the encrypted color image from the three encrypted components.
   G. Save the encrypted image.

3) Phase 3: Image decryption
   This phase can be implemented applying the following tasks:
A. Get the encrypted image.
B. Extract the color components (Red, Green, and Blue).
C. Load the key.
D. Adjust the key to suit the 2D color matrix.
E. Get the decrypted Red, Green, and Blue applying matrix multiplication of the inverse key and each one of the 2D matrices?
F. Compose the decrypted color image from the three decrypted components.

**Method 2:** Encryption using the reshaped 3D to 2D color image.

To do this we have to apply the following phases:

1) **Phase 1:** Private key generation

This phase is to be implemented one time and it includes the following tasks:

A. Generate a huge 2D matrix which is to be called key.
B. Save the key.

2) **Phase 2:** Image encryption

To do this we have to implement the following tasks:

A. Get the original image.
B. Reshape the 3D color matrix to 2D matrix
C. Load the key.
D. Adjust the key to suit the 2D matrix.
E. Get the encrypted 2D matrix by applying matrix multiplication of the key and 2D matrix.
F. Reshape the encrypted 2D image to 3D color matrix.
G. Save the encrypted image.

3) **Phase 3:** Image decryption

This phase can be implemented applying the following tasks:

A. Get the encrypted image.
B. Reshape the 3D color matrix to 2D matrix
C. Load the key.
D. Adjust the key to suit the 2D matrix.
E. Get the decrypted 2D matrix by applying matrix multiplication of the inverse of the key and 2D matrix.
F. Reshape the decrypted 2D image to 3D color matrix.
**Method 3:** Color image encryption by XORING the image with private secret key.

To do this we have to apply the following phases:

1) Phase 1: Private key generation

This phase is to be implemented one time and it includes the following tasks:

   A. Generate a huge 3D matrix which is to be used as a private key.
   B. Save the key.

2) Phase 2: Image encryption

To do this we have to implement the following tasks:

   A. Get the original image.
   B. Retrieve the image size.
   C. Load the key.
   D. Adjust the private key dimensions to match the 3D image matrix size.
   E. Get the encrypted 3D matrix by applying matrix XORING of the private key and 3D matrix.
   F. Save the encrypted image.

3) Phase 3: Image decryption

This phase can be implemented applying the following tasks:

   A. Get the encrypted image.
   B. Retrieve the image size.
   C. Load the key.
   D. Adjust the private key dimensions to match the 3D image matrix size.
   E. Get the decrypted 3D matrix by applying matrix XORING of the private key and 3D matrix.

**Method 4:** The proposed method

Color image encryption by applying XORING operation between the image blocks with private secret key.

To do this we have to apply the following phases:

1) Phase 1: Private key generation

This phase is to be implemented one time and it includes the following tasks:

   A. Generate a huge 1D array which is to be used as a private key.
   B. Save the key.
C. Generate a block (window) size.
D. Save the window size.

2) Phase 2: Image encryption

To do this we have to implement the following tasks:

A. Get the original image.
B. Reshape the 3D image to 1D array.
C. Load the key.
G. Adjust the private key dimensions to match the 1D image matrix size.
H. Get the window size
I. Get the encrypted 1D matrix by applying XORING of the private key and each block in the 1D array.
J. Reshape 1D array back to 3D matrix to form the encrypted color image
K. Save the encrypted image.

3) Phase 3: Image decryption

This phase can be implemented applying the following tasks:

A. Get the decrypted image.
B. Reshape the 3D image to 1D array.
C. Load the key.
D. Adjust the private key dimensions to match the 1D image matrix size.
E. Load the window size
F. Get the decrypted 1D matrix by applying XORING of the private key and each block in the 1D array.
G. Reshape 1D array back to 3D matrix to form the decrypted color image

3- Implementation and Experimental Results

Diffident color images in sizes and types were implemented applying each of the above mentioned methods of color image encryption-decryption, a matlab code was written for each method, and the codes were implemented using the selected images.

Encryption and decryption times were calculated by the programs.

Table (1) and (2) show the experimental results:
Table (1): Results for methods 1 and 2

<table>
<thead>
<tr>
<th>Color image size(Mbyte)</th>
<th>Method1: Encryption each color alone(matrix multiplication)</th>
<th>Method2: Encryption reshaped color image to 2D matrix(matrix multiplication)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Encryption time(seconds)</td>
<td>Decryption time(seconds)</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>0.1345</td>
<td>0.0250</td>
<td>0.0630</td>
</tr>
<tr>
<td>0.1440</td>
<td>0.0260</td>
<td>0.0650</td>
</tr>
<tr>
<td>0.1443</td>
<td>0.0290</td>
<td>0.0780</td>
</tr>
<tr>
<td>1.8024</td>
<td>1.1630</td>
<td>3.3100</td>
</tr>
<tr>
<td>2.3848</td>
<td>1.4690</td>
<td>3.6910</td>
</tr>
<tr>
<td>3.6403</td>
<td>3.2520</td>
<td>9.2910</td>
</tr>
<tr>
<td>5.8358</td>
<td>6.0160</td>
<td>16.1220</td>
</tr>
<tr>
<td><strong>Average=2.0123</strong></td>
<td><strong>1.7114</strong></td>
<td><strong>4.6600</strong></td>
</tr>
</tbody>
</table>

Table (2): Results for methods 3 and 4

<table>
<thead>
<tr>
<th>Color image size(Mbyte)</th>
<th>Method3: Xoring image with secret key</th>
<th>Method4: Xoring image blocks with secret key</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Encryption time(seconds)</td>
<td>Decryption time(seconds)</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>0.1345</td>
<td>0.276000</td>
<td>0.275000</td>
</tr>
<tr>
<td>0.1440</td>
<td>0.272000</td>
<td>0.273000</td>
</tr>
<tr>
<td>0.1443</td>
<td>0.279000</td>
<td>0.276000</td>
</tr>
<tr>
<td>1.8024</td>
<td>0.274000</td>
<td>0.275000</td>
</tr>
<tr>
<td>2.3848</td>
<td>0.266000</td>
<td>0.272000</td>
</tr>
<tr>
<td>3.6403</td>
<td>0.269000</td>
<td>0.273000</td>
</tr>
<tr>
<td>5.8358</td>
<td>0.277000</td>
<td>0.277000</td>
</tr>
<tr>
<td><strong>Average=2.0123</strong></td>
<td><strong>0.2733</strong></td>
<td><strong>0.2744</strong></td>
</tr>
</tbody>
</table>
From these tables we can see that the proposed method gave the best efficiency parameters by decreasing both the encryption decryption times as shown in figure(7) and (8)

Figure (7): Encryption time comparisons

Figure (8): Decryption time comparisons
The proposed method gave the parameters values:

MSE between the original image and encrypted one = 1.0155e+004

PSNR between the original image and encrypted one = 18.5679

MSE between the original image and Decrypted one = 0

PSNR between the original image and Decrypted one = infinite.

Which are an excellent parameters and as shown in figures (9), (10) and (11)
Figure (11): Decrypted color image

**Conclusion**

Four methods of color image encryption-decryption were studied and all these methods gave efficient parameters. The fourth proposed method gave the best parameters by decreasing encryption-decryption time.

The proposed method gave acceptable values for MSE and PSNR, and it is highly secure because of:

- The private key has several values and the number of values is variable and changeable.
- The window size is variable and changeable.

**References**


