An Efficient System for Encrypted Image by Using Hybrid DWT-DCT Compression Algorithm

Muslim Mohsin Khudhair

Department of Computer Information Systems, College of Computer Science & Information Technology, University of Basrah, Basrah, IRAQ

E-mail: mos1970@yahoo.com

Abstract— There has been lot of development in the field of multimedia and network technologies. With the development of multimedia and network technologies, the security of multimedia system becomes most important part in the internet when the data is transmitted over the network. If encryption is not performed then there may be possibility of stealing the information. Image compression is also essential where images need to be stored, transmitted or viewed quickly and efficiently. Therefore, there is need of system where encryption is done prior to the image compression. This paper proposed an image encryption method that is operated with Arnold transform method and image compression algorithm using Daubechies wavelet transforms and discrete cosine transform that can be used efficiently to compress the encrypted image.

Keywords— Encryption, Compression, Daubechies, Permutation, DWT, DCT

I. INTRODUCTION

In recent years, compression of encrypted data has attracted considerable research interest. The traditional way of securely and efficiently transmitting redundant data is to first compress the data to reduce the redundancy, and then to encrypt the compressed data to mask its meaning. At the receiver side, the decompression and decryption operations are orderly performed to recover the original data [1]. However, in some application scenarios, a sender needs to transmit some data to a receiver and hopes to keep the information confidential to a network operator who provides the channel resource for the transmission. That means the sender should encrypt the original data and the network provider may tend to compress the encrypted data without any knowledge of the cryptographic key and the original data. At receiver side, a decoder integrating decompression and decryption functions will be used to reconstruct the original data. Hence, image security/protection from unauthorized access becomes very important [2, 3]. Image Encryption refers to converting an image to such a format, so that it becomes unreadable to unauthorized access and can be transmitted securely over the internet. Image Decryption means to convert the unreadable format of an image to an original image. An image compression system consists of processes leading to compact representation of an image, so as to reduce total storage/transmission requirements [4].

This paper is a step forward in this regards. The rest of this paper is organized as follows: section 2 contains comprehensive clarification of the proposed system; the results for the proposed system and the conclusions, is given in sections 3 and 4, respectively.
II. COMPRESSION AND DECOMPRESSION OF ENCRYPTED IMAGE

In the proposed work, image encryption is performed with Arnold transformation method and compression-decompression by hybrid Discrete Wavelet Transformation (DWT) and Discrete Cosine Transformation (DCT). At the final, the decryption process is achieved.

As shown in Fig. 1, the input image is considered as \( I \), encryption over \( I \) is implemented using Arnold transformation method. The result obtained after encryption is considered as \( I_{\text{enc}} \) and then Daubechies Wavelet Transformation followed by a Discrete Cosine Transformation (DCT) technique is applied for compression followed by decompression process. The output after decompression has been stored as image \( I_{\text{dec}} \). Then the image \( I_{\text{dec}} \) is decrypted after decompression. The resultant image \( I_{\text{rec}} \) is evaluated using various performance parameters like MSE (Mean Square Error), PSNR (Peak Signal to Noise Ratio), CR (Compression Ratio) and compared with the results of other systems [5, 6].

![Diagram](image)

Fig. 2 General flowchart of proposed system for encryption-compression image.

A. Image Encryption

The permutation techniques are very useful in the encryption process, because the advantages of using the permutation in cryptography (simple implementation speed, and universality for most image formats). The permutations will not change the coefficients values but their locations [7]. A permutation (rearrangement) can be described by assigning successive number to the objects to be permuted and then giving the order of the objects after the permutation is applied [8].

Arnold transform is commonly known as cat face transform. Apply Arnold transform in digital image, so it can change the layout of gray values by change the coordinates of pixels. Seen the digital image as a \( N \times N \) matrix, then can achieve image pixels scrambling by formula [9]:

\[
\begin{bmatrix}
X_n \\
Y_n
\end{bmatrix} = \begin{bmatrix}
1 & 1 \\
1 & 2
\end{bmatrix} \times \begin{bmatrix}
X \\
Y
\end{bmatrix} \mod N
\]

(1)

Where \((X,Y)\) is the location coordinates of the original image pixels, and \((X_n,Y_n)\) is the location coordinates of image pixel that after transform [10]. Fig. 2 shows the encryption image after applied Arnold transform over original image.

![Images](image)

(a) Original image Lena, (b) Encrypted image.
B. Hybrid DWT-DCT compression algorithm

The main objective of the presented hybrid DWT-DCT algorithm for image compression is to exploit the properties of both the DWT and the DCT (as illustrated in Fig. 3). The input image is first decomposed using DWT and then DCT is used to further enhance the compression performance.

1) Discrete Wavelet Transform

Discrete Wavelet transform (DWT) is the first phase in the proposed image compression algorithm, to produce four sub-bands (See Fig. 3). The top-left corner is called LL, represents low-frequency coefficients, and the top-right called HL consists of residual vertical frequencies. The bottom-left corner LH, and bottom-right corner HH are represents residual horizontal and residual vertical frequencies respectively. The high-frequency components (LH1, HL1 and HH1) are zero or insignificant. This reflects the fact that much of the important information is contained in the LL sub-band. For this reason all the high frequency domains are discarded in this research (i.e. set all values to zero)\[11, 12\]. In particular, the Daubechies wavelet transform has the ability to reconstruct approximately the original image. This property allows higher compression ratios; this is because high frequencies from the first level can be ignored without loss of accuracy\[13, 14\].

2) Discrete Cosine Transform

The LL sub-band partitioned into non-overlapped 8×8 blocks, each block is transformed by using two-dimensional DCT to produce de-correlated coefficients. Each 8×8 frequency domain consists of: DC-value at the first location, while other coefficients are called AC coefficients\[15, 16\]. The main reason for using DCT to split the LL sub-band into two different matrices is to produce a low frequency matrix (DC-Matrix) and a high-frequency matrix (AC-Matrix). After applying the two-dimensional DCT on each 8×8 block, each block quantized by the Quantization Matrix (QM) using dot-division-matrix with truncates the results. This process removes insignificant coefficients and increases the number of the zeros in each block\[17, 18\]. QM computes as follows:

\[
QM(i, j) = \begin{cases} 
\text{Block} + (i, j), & \text{if } (QM(i, j)) = \text{odd} \\
\text{Block} + (i, j) + 1 & \text{if } (QM(i, j)) = \text{even}
\end{cases}
\]  (2)

Where Block is represented block size and \(i, j=1,2,3,..., \text{Block}\).

\[QM(i, j) = QM(I, j) \times Scale\]  (3)

In the above Equation (3), the factor (Scale) it is used to increase/decrease the values of the QM. Thus, image details is reduced in case of the factor Scale > 1. There is no limited range for this factor, because this is depends on the DCT coefficients.

© 2016, UCSMC All Rights Reserved

95
Each quantized $8 \times 8$ block is converted into one-dimensional array (i.e. the array contains 64 coefficients) by zigzag scan [14]. Whereas, the first value transferred into new array called DC-Array, while others are 63 coefficients are stored to new matrix $LL_{AC}$. Finally, the DC-Array is compressed by using Arithmetic coding. The Arithmetic coding is one of the important methods used in data compression method, especially this method used in JPEG2000. Arithmetic coding depends on (Low) and (High) equations to generate streams of bits [18].

Minimize-Matrix-Size Algorithm uses for coding $LL_{AC}$ Matrix. This algorithm applied on each three coefficients, to produce single data. This means reduce each three columns to single coded array which is called Minimized-Array. However, the bit size for each data in the Minimized-Array increased. Fig. 4 shows converting three columns into one dimensional array [4, 19]. In above Fig. 4 (a) K1, K2 and K3 represents key for conversion. The following equation illustrates converting three data, to single data.

$$D_i = (K_1 \times A_i) + (K_2 \times B_i) + (K_3 \times C_i)$$  \hspace{1cm} (4)

Where $i=1,2,3...,n$.

The key values generated through random number generator and uses in coding and decoding. For example, assume we have the following array: [3 -9 0], Maximum value in the array=|-9|=M= 9, and Base Value=0.1; Key1= 0.8147, Key2=0.9058, and Key3=0.1270. The key generated once for all matrix data, after calculation, all coded data ($D_i$) arranged together to be one-dimensional array (i.e. Minimized-Array). Before apply the Minimize-Matrix-Size algorithm, the algorithm computes the probability of the data for AC-matrix. These probabilities are called Limited-Data, which is used later in decompression stage [14]. The Limited-Data stored as additional information with compressed data. Fig. 5 describes Limited-Data computed from original matrix.

The Minimized-array contains positive and negative data, and each data size reached to 32-bit, these data can be compress by a coding method, but the index size reaches to 50% of compressed data size. The index data are used in decompression stage, therefore, the index data breakup into parts for easy compress. Each data partitioned into parts: 4-bit (i.e. each data in index may be breakup into six 4-bit data), and this process increase the probability of redundant data.

The final step of the compression algorithm is arithmetic coding, which is one of the important methods used in data compression; its takes a stream of data and convert it to a single floating point value. These values lie in the range less than one and greater than zero that, when decoded, return the exact stream of data. The arithmetic coding needs to compute the probability of all data and assign a range for each data (low and high) in order to generate streams of bits [18, 20].
C. Decompression algorithm

The decompression algorithm represents reverse steps for the proposed image compression. Firstly, applied arithmetic decoding for decompress DC-array, Nonzero-array and Zero-array. Thereafter, nonzero-array and zero-array are combined together for reconstructing minimized-array. Secondly, using Parallel Sequential Search Algorithm (PSS-Algorithm), moreover, this algorithm represents inverse Minimize-Matrix-Size Algorithm for reconstructing AC-Array. PSS-Algorithm, estimates (Ai, Bi and Ci) by using (Di) with Key. Whereas, Ai, Bi and Ci represents estimated columns for decompress AC-Matrix [17, 20, 21]. PSS-Algorithm can be illustrates in the following steps:

Step 1: PSS-Algorithm starts to pick first (P) data from the Limited-Data, and then these (P) data are connected with each other look like a network as shown in Fig. 6. In Fig. 6 (Column-1) data connected as a network with (Column-2) data, also (Column-2) is networking with (Column-3). In another words, the searching algorithm computes all options in parallel. For example: A=[1 -1 0], B=[1 -2 0] and C=[3 -1 5], and P=3, according to Equation (4) (A), (B) and (C) computes 27 times. This means, all options computes in parallel and one option will be matched with the (Di), and (Ai), (Bi) and (Ci) in (Column-1), (Column-2) and (Column-3) represented decompressed data.

Initially, PSS algorithm starts with P=10 data from (Limited-Data(1...10)) that used by the algorithm, these data are estimates three columns (A, B and C), as mentioned in Fig. 6 (a). Thereafter, the algorithm starts searching for original data (Ai, Bi and Ci) which is depends on compressed column (Di) and Key-values. The first iteration for the algorithm starts with matching selected (Di) with 10 outputs from PSS-algorithm (i.e. P=10, three columns = P3= 1000 data). In another words, Equation (4) executed 1000 times in parallel for finding original values for columns (A, B and C) as mentioned in Fig. 6 (b). If result unmatched, in this case (Array2=Array1) (i.e. transferred data from Array1 to Array2), then new processing starts. This process will continue until finding all original columns (Ai, Bi and Ci) in AC-Matrix.
Step 2: In this step decompressed AC-Matrix composed with each DC-value (i.e. DC-values from DC-array), then followed by zigzag scan to convert each 64-coefficients to $8 \times 8$ blocks. These blocks combined with each other to build LL sub-bands. Subsequently, applied inverse quantization (i.e. dot-multiplication), followed by inverse DCT on each $8 \times 8$ block. Finally, applied inverse DWT for obtaining 2D image. The decompression algorithm steps are showed in Fig. 7.

D. Image Decryption

At the receiver side, the original image can be reproduced by the inverse permutation (Arnold transform) with reverse order of this process as shown in Fig. 8.
III. RESULTS AND DISCUSSIONS

The current technique is applied on six grayscale images labelled image 1 to image 6 as shown in Fig. 9. Each of these images has size of (256×256 pixels) with 8-bit grey levels. It was implemented with (MATLAB 2012) package. The implementation was done on a PC (DELL laptop) with 2.1 GHz core 2 due processor and 2GB main memory running with windows 7 operating system.

The proposed work is analyzed by using various parameters like MSE (Mean Square Error), PSNR (Peak Signal to Noise Ratio) and CR (Compression Ratio) [22].

The mean square error (MSE) is used as metric to measure the distortion between original and reconstructed image [23]. The equation that evaluating the MSE is:

\[
MSE = \frac{1}{M \times N} \sum_{x=1}^{N} \sum_{y=1}^{M} [I(x, y) - I_{rec}(x, y)]^2,
\]

where:

- \( I \) : is the original image.
- \( I_{rec} \) : is the reconstructed image.

Fig. 8 (a) Decompressed image, (b) Decrypted image.

Fig. 9  Test Images
The peak signal to noise ratio (PSNR), in decibels (dB), can be evaluated as follows [24]:

\[
PSNR = 10 \log_{10} \left( \frac{P_{\text{in}}^2}{MSE} \right),
\]

where \( P_{\text{in}} \) is maximum possible pixel value, e.g. 255 in an 8-bit grey-level image.

Compression Ratio (CR) refers to ratio of the size of an original image to the size of reconstructed image. It is defined as follows [22]:

\[
CR = \frac{I}{I_{rec}}
\]

In our work, we evaluate different systems as described below:

- **System1** uses our propose work.
- **System2** uses standalone DCT in compression-decompression stage to encrypted image.
- **System3** uses standalone DWT in compression-decompression stage to encrypted image.

To test performance our system compared with other systems, we use different criteria such as MSE, PSNR and CR. The values of PSNR and MSE are compared in different systems, as shown in Table 1.

### Table 1
COMPARISION OF PSNR AND MSE FOR SYSTEM1, SYSTEM2 AND SYSTEM3.

<table>
<thead>
<tr>
<th>Image No.</th>
<th>System1</th>
<th>System2</th>
<th>System3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR</td>
<td>MSE</td>
<td>PSNR</td>
</tr>
<tr>
<td>1</td>
<td>37.1091</td>
<td>12.4722</td>
<td>35.9735</td>
</tr>
<tr>
<td>2</td>
<td>37.3653</td>
<td>12.1568</td>
<td>35.8653</td>
</tr>
<tr>
<td>3</td>
<td>36.8365</td>
<td>13.5778</td>
<td>35.7074</td>
</tr>
<tr>
<td>4</td>
<td>37.4139</td>
<td>12.0979</td>
<td>36.0513</td>
</tr>
<tr>
<td>5</td>
<td>36.8370</td>
<td>13.5760</td>
<td>35.8070</td>
</tr>
<tr>
<td>6</td>
<td>35.3995</td>
<td>18.9032</td>
<td>34.3864</td>
</tr>
</tbody>
</table>

Fig 10 shows the PSNR values of six encrypted images for average compression ratio of 96% by System1, System2 and System3.
Similarly, Fig. 10 shows the compression ratio of six encrypted images for average PSNR of 32 db, when compressed by System1, System2 and System3.

![Fig. 10 PSNR of images for System1, System2 and System3.](image)

![Fig. 11 CR of images for System1, System2 and System3.](image)
IV. CONCLUSIONS AND FUTURE WORKS

In this paper, an efficient image encryption and compression system is designed using image transformation. An From the experimental results it is evident that, the proposed system technique gives better performance compared to other traditional systems techniques. In another meaning, the value of PSNR, CR are high and the value of MSE is low that means our technique is effective. Also the results demonstrate that for test images, the loss of information is less hence the quality is better. In future, the technique can be extended by applying different transforms on color image. High performance compression algorithms may be developed and implemented using neural networks and soft computing.

REFERENCES


