Compression of Visual Cryptography Shares Authenticated using QR Code

1Mary Shanthi Rani, M1,2, Germine Mary, G.
1Department of Computer Science and Applications, Gandhigram Rural Institute-Deemed University, Gandhigram, India
2Department of Computer Science, Fatima College, Madurai, India

Abstract—Visual Cryptography is a Secret Image Sharing scheme which conceals a secret image into two or more images called shares. These shares are meaningless images and look like random dots. The secret image is recovered by stacking the shares. This paper proposes a novel method to authenticate so that only shares from authorized users are combined to reveal the secret image. This is done by first embedding QR Code (abbreviated from Quick Response Code) into the secret image, followed by the creation of shares. The original secret image is not affected as we embed the QR Code in non-region of interest. If a modified share or unauthorized share is superimposed on to the original share, the embedded QR Code cannot be read/retrieved from the resultant image. This signifies that one of the shares is not an authorized one. In VC the reconstructed image, encounter a major problem of pixel expansion. This is also solved in the proposed method by compressing the shares using a hybrid method that combines Vector Quantization and Run Length Encoding methods to achieve lossless compression. Decompression is done by applying decoding procedure and the shares are overlapped to view the secret image.

Keywords— Image Compression, QR Code, Run Length Encoding, Vector Quantization, Visual Cryptography

I. INTRODUCTION

The Visual Cryptography (VC) was first raised by Naor and Shamir in 1994 [1]. VC is a powerful method of Information Security, which indicates the possibility of visually protecting vital secrets from the view of secret sharing. Unlike generally used security methods, which tend to hide information by applying a mathematical transformation on secret; Visual Cryptography Scheme (VCS) stores the secret as an image. In VCS, secret image (SI) is split into several images called VC shares, where every piece of VC shares has no hint of the original SI, while receivers can clearly recognize the secret by simply overlaying these shares. Certain qualified subsets of participants can “visually” recover the SI, but other, forbidden, sets of participants have no information on SI. Hence, VC is a desirable security scheme for the safety of confidential documents. VCS depends on the perception of Human Visual System (HVS) [1, 2]. On superimposed VC secret image, groups of black and white sub-pixels represent light regions while dark regions are filled with only black sub-pixels. The contrast of the light regions and dark regions can be easily identified by HVS.

The main advantages of VC are:

- VC secret is revealed by using only VC shares, which are suitable to carry with, unlike other cryptography methods that require complex computations and powerful computers for decryption.
- VC concentrates on using images as the media for transmission of secret information in contrast to the cryptographic methods that deal with a string of characters or files. Compared with plain text encryption, an image is flexible in conveying secret content and has larger information storage.
- VC is a one-time padding cryptographic technique, which makes it unconditionally secure.

However, the advantages of VC are high for the relevant security applications; previous researches have reported the existence of authentication problem [3].

Generally, the VC shares are sent to different persons and they come together to overlap the shares to view the secret. There is a possibility of mixing up of the genuine shares and the duplicate shares or shares of another secret message, as all shares look like random dots or noise. Therefore, the security, identification, and protection of genuine VC shares are mandatory to make sure the successful revealing of a secret. Previous studies reveal that participants in VC lack the ability to identify the authenticity of all shares and the secret, hence giving unauthorized persons an opportunity to create unauthorized shares stacking which may give an undesirable result.

This problem of authentication is overcome by embedding QR Code into the SI at an appropriate corner without damaging the original image. VC shares are created and sent as usual to multiple participants. When it is overlapped, QR Code can be read through mobile to confirm whether all the received shares are genuine. Even if one share is modified / distorted then QR Code could not be read. Embedding of QR code not only improves the authentication, it also helps to hide additional information.

This paper includes specific VC authentication method by embedding QR Code using image segmentation at the non-region of interest of an image. There are five main benefits of using QR Codes in authentication [4].
Instead of using a share for authentication, QR code can be embedded into shares, thereby simplifying the authentication process. In addition, cheaters can hardly get the information of the QR code from the prediction of a secret. As the QR code is able to represent a large volume of data within a small size of the pattern, it can be used as a tool to transport additional secret. QR-Code also has error correction capability. Data can be restored even when substantial parts of the code are distorted or damaged. As many applications on mobile devices and personal computers have been developed for scanning QR code, it is convenient for users to decode the QR code by using the built-in cameras on the users’ cell phones or laptops.

Image Compression is the process of reducing the number of bits required to represent an image. Compression has traditionally been done with little regard for image processing operations that may precede or follow the compression steps. In this proposed scheme, compression follows VC as it results in pixel expansion [5]. Data compression is the mapping of a data set into a bit stream to decrease the number of bits required to represent the data set. With data compression, one can store more information in a given storage space and transmit information faster over communication channels. Strategies for compression are reducing redundancies and exploiting the characteristics of human vision. There are two types of data compression, lossless and lossy. Lossless compression has an advantage that the original information can be recovered exactly from the compressed data. However, lossless compression methods cannot achieve higher compression ratio. To achieve high compression ratio without compromising the quality of the image, the proposed system employs multistage approach combining lossy compression method Vector Quantization followed by Run Length Encoding.

The main objectives of this proposed method are to formulate a secret sharing system, which has the following characteristics.

- To embed the QR code into the SI to improve the authentication.
- Exploit the advantage of VC to create meaningless shares to hide a secret.
- Minimizing the drawback of VC, that is image expansion, by applying multistage Image Compression.

Section 2 gives literature review of VC creation, QR Code and Image compression techniques used in this proposed method. Section 3 introduces the architecture and system design of the proposed method. Section 4 describes the experimental results and outcomes. Section 5 presents the analysis and discussion of the results. Section 6 concludes highlighting the application, restrictions and future enhancement of this method.

II. LITERATURE REVIEW

A. Introduction to VC

In a (2, 2) - threshold visual secret sharing scheme, let the SI be a binary image with size \(N \times N\). To begin with, every pixel is extended into a 2x2 block, and each block is composed of two black pixels and two white pixels as shown in Fig. 1. We use pixel value of ‘1’ and ‘0’ representing black and white pixels respectively. Constructing the pixels of two basis matrices \(C_0\) and \(C_1\) is indicated below.

\[
C_0 = \{ \text{all the matrices obtained by permuting the columns of } \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix} \}
\]

\[
C_1 = \{ \text{all the matrices obtained by permuting the columns of } \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \}
\]

where \(C_0\) is used to represent shares of white pixel and \(C_1\) is used to represent shares of a black pixel.

For example in Fig. 2, the SI (a) is decomposed into two visual cryptography transparencies (b) and (c). When stacking the two transparencies, the reconstructed image (d) is obtained.

![Fig. 1 Sharing and stacking scheme of the black and white pixel](image-url)
Even though the contrast of the resulting image is degraded by 50%, human eyes can still identify the content of the SI. SI is shown only when both shares are superimposed. In VC, the contrast is evaluated by using Hamming weight $H(V)$ of the ORed $m$-vector $V$ is interpreted by the visual system as follows:

A black pixel is interpreted if $H(V) \leq d$ and white if $H(V) < d - \alpha m$ for some fixed threshold $1 \leq d \leq m$ and a relative difference $\alpha > 0$.[6,7]

**B. Introduction of QR Code**

The QR Code [8] is a two-dimensional barcode introduced by the Japanese company Denso-Wave in 1994. It contains information in both vertical and horizontal directions, whereas a classical barcode has only one direction of data. Compared to the classical barcode, a QR-code can hold a considerably greater volume of information, 7,089 characters for numeric only, 4,296 characters for alphanumeric data, and 2,953 bytes of binary (8 bits). The “QR” is derived from “Quick Response”, as the creator intended the code to allow its contents to be decoded at high speed. In addition, QR-Code also has error correction capability. Data can be restored even when substantial parts of the code are distorted or damaged. The QR Code is capable of high-speed reading in all directions (360°) from three corners implemented with the finder patterns to notify the position of the code. Many cellular phones with embedded camera nowadays are natively equipped with the QR-code decoding software. With the aid of the equipment, it is simple for a human to manually decode QR-codes and then to display, manipulate, or store the information on their mobile devices.

**C. Image Segmentation and Region Growing**

The main goal of segmentation is to partition an image into a set of disjoint regions that are visually different, homogeneous and meaningful with respect to some characteristics or computed property such as grey level, texture or color to enable easy image analysis. Visually it is very easy to determine what a region of interest is and what is not. Doing so with a computer algorithm, on the other hand, is not so easy [9]. Region Growing is an approach to image segmentation in which neighboring pixels are examined and added to a region class if no edges are detected. This process is iterated for each boundary pixel in the region. The region-based segmentation is the partitioning of an image into similar/homogenous areas of connected pixels through the application of homogeneity/similarity criteria among candidate sets of pixels.

**D. Vector Quantization**

Vector Quantization (VQ) is a popular image compression algorithm for reducing the transmission bit rate or storage, which maps the pixel intensity vectors into a set of a limited number of code vectors. A block-coding technique quantizes blocks of data instead of a single sample. VQ exploits the correlation between neighboring signal samples by quantizing them together [5, 10].
VQ Compression contains two components: VQ encoder and decoder as shown in fig. 4. At the encoder, the input image is partitioned into a set of non-overlapping image blocks. The closest codeword in the codebook is then found for each image block. Next, the corresponding index for each searched closest code word is transmitted to the decoder. Compression is achieved because the indices of the closest code words in the codebook are sent to the decoder instead of the image blocks themselves.

E. Run Length Encoding

In an image matrix, once a pixel takes on a particular color (black or white), it is highly likely that the following pixels will also be of the same color. Therefore, rather than code the color of each pixel separately, we can simply code the length of the runs of each color. For example, if we had 190 white pixels followed by 30 black pixels, followed by another 210 white pixels, instead of coding the 430 pixels individually, we would code the sequence 190_30_210, along with an indication of the color of the first string of pixels. Coding the lengths of runs instead of coding individual values is called Run-Length Encoding [5].

III. PROPOSED METHOD

The main objectives of this proposed method is to formulate a secret sharing system that has the following objectives:

- Exploit the advantage of VC to create meaningless shares to hide SI.
- Enhance the authentication of VC shares using QR Codes and to hide additional secret message using QR Codes.
- Minimizing the drawback of VC, that is image expansion, by applying multistage Image Compression.

The proposed system consists of five phases. The first phase focuses on finding the non-region of interest (NROI), i.e., the corner with the maximum background. In the second phase, QR code is embedded in NROI of SI obtained from the previous phase. In the third phase, general VC scheme is applied to the embedded SI to create 2 shares. The size of the shares will be 4 times the size of the original SI as each pixel is replaced by a 2X2 block of the pixel as illustrated in fig. 6. Shares generated in the third phase are compressed using VQ followed by RLE methods and the images are converted to few bits in the fourth phase. Decompression is done in the final phase and the shares are overlapped to view the SI and the secret message.

A. Phase I: Image segmentation with region growing technique to locate a corner in the SI with the maximum background to embed QR Code

Previous research work shows that the region of interest will be generally concentrated at the center of the image and the corners are commonly filled with background [11]. This advantage is taken into consideration in the proposed method. A novel image segmentation using region-growing technique is used to find the corner in SI that has a maximum background color so that QR code can be embedded in that region. The steps involved in this phase are given below:

Step 1: Convert the given image into a binary image.

Step 2: Split the image into block of pixels following the steps below:

a) Calculate the width (w) and height (h) of the given SI.

b) For the given w and h find, all the factors and store it in two arrays aw and ah.

c) \[ n = \frac{(w \times h)}{(aw(i) \times ah(j))} \]  

(1)

where \( n \) represents the number of blocks and the denominator determine the block size.

d) Example : if \( w = 100, \ h = 120 \) and \( n = 1000 \) then 

\[ aw = 2, 4, 5, 10 \] \[ ah = 2, 3, 4, 5, 8, 10, 12, 15, 24, 30, 40, 60 \] 

substituting in (1) we get the denominator \( 12 \), indicating the value of I and j to be equal to 2 i.e. \( aw(i) = 4 \) and \( ah(j) = 3 \).
Step 3: Initialize the algorithm by selecting the initial block \( b_i \), that is any one corner of the image as shown in Fig. 5, where each square represents a block i.e., group of pixels and find the region with maximum background using the following steps:

a) If the sum of the background color in that block is > 80% include the block in the region and consider the next block in the row to include in the region until the block background intensity is > 80%. The region \( R_1 \) represents the total blocks of pixels in one corner including only row-wise blocks.

\[ R_1 = \sum_{i=1}^{b} b_i \quad \text{(where i=i+1)} \]

b) \[ R_1 = \sum_{i=1}^{b} b_i \quad \text{(2)} \]

c) Continue with \( b_i \), to include blocks in the next column and check its background intensity to extend the region. Combine this region with the previous step and we get the region \( R_1_{rc} \).

d) \[ R_1_{rc} = \sum_{i=1}^{b} b_i \quad \text{(where i=i+1)} \]

e) Repeat the previous step for the diagonal blocks and add it to the region to get \( R_1_{rkd} \).

\[ R_1_{rkd} = \sum_{i=1}^{b} b_i \quad \text{(where i=i+bw +1)} \]

f) \[ R_1_{rkd} = \sum_{i=1}^{b} b_i \quad \text{(3)} \]

g) \[ R_1_{rkd} = R_1 = (2) U (3) U (4) \]

Step 4: The above steps are repeated for all the four corners to get \( R_2_{rkd} \), \( R_3_{rkd} \), and \( R_4_{rkd} \) as follows:

\[ R_2_{r} = \sum_{i=1}^{b} b_i \quad \text{(where i = i+bw)}, \]

\[ R_2_{c} = \sum_{i=1}^{b} b_i \quad \text{(where i = i-1)}, \]

\[ R_2_{d} = \sum_{i=1}^{b} b_i \quad \text{(where i = i+bw-1)}, \]

\[ R_3_{r} = \sum_{i=1}^{b} b_i \quad \text{(where i = i-bw)}, \]

\[ R_3_{c} = \sum_{i=1}^{b} b_i \quad \text{(where i = i+1)}, \]

\[ R_3_{d} = \sum_{i=1}^{b} b_i \quad \text{(where i = i-bw+1)}, \]

\[ R_4_{r} = \sum_{i=1}^{b} b_i \quad \text{(where i = i-bw)}, \]

\[ R_4_{c} = \sum_{i=1}^{b} b_i \quad \text{(where i = i-1)}, \]

\[ R_4_{d} = \sum_{i=1}^{b} b_i \quad \text{(where i = i-bw-1)}. \]

Step 5: The max (\( R_i \)) where i = 1 to 4 is the non region of interest which represents the corner with maximum background color.

Step 6: The value of i indicates the corner to embed the QR code and \( b_i \) the size of the region.

**B. Phase II: Embed QR Code in the SI**

In this phase, QR code is embedded into one corner of the SI which has a maximum background color so that the SI remains intact. The size of the QR code is compared with the size of the region obtained in the previous phase and the size of QR code is modified if needed. QR Code can be resized to 50% without loss of data.

Step 1: Get the output of phase 1 and determine the size of the NROI (background).

Step 2: Read the QR Code and determine its size.

Step 3: Compare the size derived from step 1 and step 2. If step 2 value Is greater than step 1 value then resize the QR Code image.

Step 4: Embed the QR Code into the SI.

**C. Phase III: Creation of VC Shares**

The SI sharing scheme proposed by Naor and Shamir is used in this phase, in which a secret binary image that has pixels with only two values (black/white) is encrypted into two shares. If the pixel is white, one of the two rows of the white pixel as shown in Fig. 6(b) is chosen to generate two shares. Similarly, if the pixel is black, one of the lower two
rows is chosen to generate the two shares. Fig. 6(b) shows the possible values of pixels in each of the two generated shares. The reconstruction of the SI is simply done by stacking the pixels again as shown in Fig. 6(b). There is an expansion of pixels in the shares and the expansion factor is four, which is the block size used to replace a single pixel. This drawback is overcome by reducing the share size using Phase IV.

![Fig. 6 Phase III – Creation of VC Shares.](image)

**D. Phase IV: Image Compression (Encoding)**

VQ technique is applied to the shares generated in phase III. The basic idea behind this technique is to develop a dictionary of fixed-size vectors, called code vectors (Code Book). A vector is usually a block of pixel values. A given image is then partitioned into non-overlapping blocks (vectors) called image vectors of size 4. Then each image vector is compared with the code vectors in the dictionary and its index in the dictionary is determined which is used to encode the original image vector. Thus, each image is represented by a sequence of indices and is stored in a Vector Index Table (VIT) as shown in Fig. 7 [12].

The size of VIT is further reduced by applying RLE algorithm. This is a very simple compression method used for sequential repetitive data. This technique replaces sequences of identical symbols (index value in our case), called runs in a separate vector. The Code Index in VIT has just two values 0 or 1 to represent the index of code vector. This is passed as an input to RLE algorithm. RLE creates two vectors, one to represent the arrangement of the element and the other to represent the runs of the element. These two vectors and the code vector is the final output sent to the receiver. The phase II – encoding process is shown in Fig. 7.

**Image Compression algorithm:**

1. Step 1: Read the shares and perform the following steps for both the shares.
2. Step 2: Convert the image into blocks of size 2X2 and then to a vector of size 4.
3. Step 3: Compare each vector with the predefined vector used for creating shares. The Codebook has 2 vectors (1, 0, 1, 0) and (0, 1, 0, 1).
4. Step 4: Create Vector Index Table (VIT) that contains block number and codebook index.
5. Step 5: Code Index in VIT has a sequence of two values 0 or 1 to represent the index of the codebook. This is passed as an input to RLE algorithm.
6. Step 6: RLE creates two vectors, one to represent the arrangement of the element (code indices) and the other to represent the no. of occurrences of each element. These two vectors and the codebook is the final output sent to the receiver.

![Fig. 7 Phase IV – Image Compression](image)

**E. Phase V: Image Decompression (Decoding)**

The decoding process of the compressed bit stream is done in Phase V to reveal the SI. Using the two vectors, elements and run vectors VIT is recreated using the reverse algorithm. Similarly, using VIT and Code Vectors the Image Vectors (blocks) are formed which are combined together and the shares are created. By simply overlapping the shares, SI can be seen using our HVS. The Phase V decoding process is illustrated in fig. 8.
Decompression and Revealing SI:
Step 1: From the two vectors (element and length) VIT is recreated using the reverse algorithm.
Step 2: Using VIT and Code book, the blocks and finally the shares are created.
Step 3: By simply overlapping the shares SI can be seen using our HVS.

Fig. 8 Phase V – Image Decompression

IV. EXPERIMENTAL RESULTS

Using the above-proposed method experiments were conducted on several binary test images of size 512 x 512 pixels and 2 x 2 pixel block size using Pentium Dual-Core processor @ 2.5 GHz with 2GB RAM. The results of all the five phases are given below. Phase2 of the suggested method is tested by embedding QR codes with messages ranging from 4 bytes to 100 bytes. MSE and PSNR are evaluated between original secret image and QR Code embedded the image and is tabulated in Table II. It is observed that the PSNR value ranges from 15.86 to 16.17. PSNR, Q, Bits per pixel (bpp), Compression / Decompression time and Compression ratio values for standard secret images Baboon, Lena and Peppers for QR codes with information ranging from 4 bytes to 100 bytes is given in Table III, IV and V respectively.

Table I Result of Phase I, II and III

Table II Comparison of Images Using MSE and PSNR

<table>
<thead>
<tr>
<th>QR Code</th>
<th>Secret Image</th>
<th>SI with embedded QR Code</th>
<th>VC Share1</th>
<th>VC Share2</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="QR Code" /></td>
<td><img src="image2" alt="Secret Image" /></td>
<td><img src="image3" alt="SI with embedded QR Code" /></td>
<td><img src="image4" alt="VC Share1" /></td>
<td><img src="image5" alt="VC Share2" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Original Secret Image</th>
<th>Original message in QR Code</th>
<th>Data in bytes</th>
<th>QR Code</th>
<th>QR Embedded</th>
<th>MSE</th>
<th>PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>abcde</td>
<td>4</td>
<td>1686.53</td>
<td>15.86</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trust in the LORD</td>
<td>17</td>
<td>1655.71</td>
<td>15.56</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Secret message hidden</td>
<td>32</td>
<td>1626.09</td>
<td>18.02</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Although initially used</td>
<td>97</td>
<td>1569.84</td>
<td>16.17</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Failure will never overtake me if my determination to succeed is strong enough. A.P.J Abdul Kalam</td>
<td>160</td>
<td>1569.85</td>
<td>16.17</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table III Performance Metrics of Compression of Baboon Image

<table>
<thead>
<tr>
<th>Image</th>
<th>Performance Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>QR Code</td>
</tr>
<tr>
<td><img src="image1.png" alt="Baboon Image" /></td>
<td><img src="image2.png" alt="Baboon Image" /></td>
</tr>
<tr>
<td><img src="image11.png" alt="Baboon Image" /></td>
<td><img src="image12.png" alt="Baboon Image" /></td>
</tr>
</tbody>
</table>

### Table IV Performance Metrics of Compression of Lena Image

<table>
<thead>
<tr>
<th>Image</th>
<th>Performance Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>QR Code</td>
</tr>
<tr>
<td><img src="image1.png" alt="Lena Image" /></td>
<td><img src="image2.png" alt="Lena Image" /></td>
</tr>
<tr>
<td><img src="image11.png" alt="Lena Image" /></td>
<td><img src="image12.png" alt="Lena Image" /></td>
</tr>
</tbody>
</table>

### Table V Performance Metrics of Compression of Peppers Image

<table>
<thead>
<tr>
<th>Image</th>
<th>Performance Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>QR Code</td>
</tr>
<tr>
<td><img src="image1.png" alt="Peppers Image" /></td>
<td><img src="image2.png" alt="Peppers Image" /></td>
</tr>
<tr>
<td><img src="image11.png" alt="Peppers Image" /></td>
<td><img src="image12.png" alt="Peppers Image" /></td>
</tr>
</tbody>
</table>
V. DISCUSSION

Image features of the original image, VC shares, and combined shares are discussed with respect to the spatial domain. The performance of this method is evaluated using standard metrics like histogram, peak signal-to-noise ratio (PSNR), and compression and decompression time, bit rate in bits per pixel, compression ratio, entropy and structured quality index (Q) to measure the quality of the reconstructed image.

Table I shows the creation of two shares from the SI. In Share1 the same 2x2 block replaces every pixel irrespective of its color as illustrated in fig. 1. (First and Third row are selected). Hence when we compress share1 it needs just 25 bits, 4 bits to represent the code vector (block) and 21 bits ($2^{20}$) to store the size of the image (1024 x 1024). So in our discussion share2 is taken into consideration for analysis.

A. Histogram

Pixel histogram is an efficient technique for representing the similarity of distribution of different pixels in the spatial domain. The vertical axis represents the number of pixels in a particular color, whereas, the horizontal axis of fig. 9 and fig. 10 represents the variations in color. The right side of the horizontal axis represents the white pixels and the left side represents black pixels. The percentage of black and white pixels in the image is represented by the vertical axis. Due to the nature of VC schemes, the ratio of black and white pixels in VC shares is approximately 1:1.

The histogram images in fig. 9(a) and 9(b) reveal that there is no much change in the histogram of the baboon image before and after embedding QR Code with 100 bytes of data. Similarly, the VC shares of baboon before and after embedding the QR Code reveals no much difference in the histogram image as shown in fig. 10(a) to 10(d).

![Histograms of baboon image](image)

**Fig. 9** Histograms (a) Baboon (b) Baboon with embedded QR code.

![Histograms of VC shares](image)

**Fig. 10** Histograms of VC shares (a) Baboon Share1 (b) Baboon Share2 (c) Baboon-QR-embedded Share1 (d) Baboon-QR-embedded Share2

B. PSNR

In statistics, the mean square error (MSE) is the measure of the average of the squares of the "errors", that is, the difference between the estimator (original image) and the estimated image (recovered image).
The peak signal-to-noise ratio (PSNR) in decibels is computed between two images. This ratio is often used as a quality measurement between the original and the reconstructed image [13]. The higher the PSNR value better is the quality of the reconstructed image. PSNR is most commonly used to measure the quality of reconstruction of lossy compression. The signal, in this case, is the original data, and the noise is the error introduced by compression.

The PSNR (in dB) is defined as:

\[
PSNR = 20 \log_{10} \left( \frac{N_{max}}{\sqrt{MSE}} \right)
\]

where 

\[
MSE = \frac{1}{m} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2
\]

Table II shows the PSNR value of the QR Embedded baboon and the original baboon image. It is interesting to note that PSNR value increases as the number of bytes hidden in QR Code increases.

The results Tables III, IV and V shows PSNR value between the compressed and reconstructed image as infinity for all images conforming the method used in this proposed method is a lossless compression.[14]

C. Bit Rate

The compression efficiency is also measured by the bit rate, which is the average number of bits required to store a pixel and is computed as follows [15].

\[
\text{Bit Rate} = \frac{C}{N} \text{ (bits per pixel) (bpp)}
\]

where C is the number of bits in the compressed file and N is the number of pixels in the original image.

The bpp values in table III to V indicate that for the same SI, bpp increases with increase in information size in QR Code. In all the standard images (Baboon, Lena, and Peppers) the bpp for QR Code with 4 bytes of information is the minimum (Baboon – 0.1849, Lena – 0.0824, Peppers – 0.0637) and QR Code with 100 bytes of information is the maximum (Baboon – 0.1936, Lena – 0.0946, Peppers – 0.076). In Table VI, the bpp for Barbara image is 0.231, which is high compared to other standard images. The bpp for cameraman is just 0.068, which shows the compression efficiency of the algorithm. From the bpp, we understand that the images with more background end up with high compression.

The total number of bits needed to transmit an original image, VC shares, and compressed shares for standard images is shown in Table VII. The table shows the efficiency of compression.

<table>
<thead>
<tr>
<th>Image</th>
<th>Performance Metrics of Compression for QR Code with 100 Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Original</td>
</tr>
</tbody>
</table>

D. Compression Ratio

Data compression ratio is the ratio of the original file size to the compressed file size. This shows how much compression is achieved for a given image and it is evident that higher compression ratio results in drastic reduction in the size of the compressed file. Table VI shows that Compression ratio of Barbara image is 34.434, whereas that of the cameraman is 116.316.

\[
\text{Compression Ratio} = \frac{\text{Uncompressed Size}}{\text{Compressed Size}}
\]  

The compression ratio (cr) values in table III to V indicate that for the same SI, cr decreases with increase in information size in QR Code. In all the standard images (Baboon, Lena, and Peppers) the cr for QR Code with 4 bytes of information is the maximum (Baboon – 43.256, Lena – 97.03, Peppers – 125.4) and QR Code with 100 bytes of information is the minimum (Baboon – 41.325, Lena – 84.48, Peppers – 105.2).
E. Compression Speed

It is the amount of time required to compress and decompress an image. This value depends on a number of factors such as the complexity of the algorithm, efficiency of software / hardware, implementation of the algorithm etc. Compression speed helps to rate the efficiency of the algorithm. The result Table VI shows the efficiency of this algorithm. The proposed method, on an average, takes 1.3 sec to compress and 0.66 sec to decompress the image [9].

F. Structured Similarity Index (Q)

A quality assessment measure for images called the Universal Image Quality Index, Q was proposed by Wang et al.[16] which is defined as

\[ Q = \frac{4 \sigma_{xy} \mu_x \mu_y}{(\sigma_x^2 + \sigma_y^2)(\mu_x^2 + \mu_y^2)} \]

where \( \mu_x \) and \( \mu_y \), \( \sigma_x \) and \( \sigma_y \) represent the mean and standard deviation of the pixels in the original image (x) and the reconstructed image (y) respectively. \( \sigma_{xy} \) represents the correlation between the original and the reconstructed images. The dynamic range of Q is \([-1, 1]\) [17]. The best value 1 is achieved for all our sample images as shown in Table VI, which shows the proposed method retains the exact original image after decompression.

G. Entropy

Entropy can be described as a measure of the amount of disorder in a system [18]. The entropy of the image decreases as the information content decreases. A full grayscale image has high entropy, a threshold binary image has low entropy and a single-valued image has zero entropy. If the pixels of an image, after inspection are found to be the same, then this information can be communicated using a very short message. If the pixels are changing in unexpected ways, then longer messages are required to communicate this fact and the information is said to increase. Entropy sets a lower bound on the average number of bits per pixel required to encode an image without distortion. Entropy is defined as

\[-\sum (p \cdot \log_2 (p)) \]

where \( p \) contains the histogram counts returned from imhist().

Table VIII The Entropy Value of Baboon Images

<table>
<thead>
<tr>
<th>Original Image</th>
<th>QR Embedded</th>
<th>VC Share1</th>
<th>VC Share2</th>
<th>Combined Share</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.184</td>
<td>1.248</td>
<td>1.0</td>
<td>1.0</td>
<td>1.556</td>
</tr>
</tbody>
</table>

The entropy value of baboon image in different stages is shown in Table VIII. The entropy of the shares is 1.0 that means an equal number of black and white pixels is present in the shares.

H. Authentication

QR Code provides authentication. It confirms that the correct shares are combined to reveal the secret. When two different shares are combined as in fig. 11, then QR code becomes unreadable and proves that the shares are not authorized ones or original shares. Moreover, more additional information can be transferred in an easy way using QR Code as it accepts nearly 4000 characters. Following screen shots illustrate the authenticity of the proposed method. Even if part of the image is cropped / resized as in fig. 12 (a), you can still read the data in QR code and retrieve some content.
VI. CONCLUSION

In this article, a new method of VC authentication is done by embedding QR Code in non-region of interest of the secret image without affecting its quality. The novelty of the proposed method is multifold. First, the secret image is authenticated by embedding QR Code. Secondly, the volume of a secret message transmitted increases in this scheme, as VC shares that hide the secret image also conceals secret message embedded in QR Code. Third, security and efficiency of transmission of VC shares are enhanced further by performing lossless compression. This paves a way to solve the problem of pixel expansion of VC shares. Though this system enhances the security of VC shares, there are few disadvantages. Unlike general VC systems that do not require any computation at the receiver, this scheme executes decompression algorithm at the receiver. Still, this is not a great issue as the decompression time takes less than a second as given in Table VI. This algorithm of embedding QR Code is only applied to (2, 2) VC. Future work can be extended to include this authentication for other types of VC schemes such as Color VC and Extended VC. Future work can be done on diverse color spaces such as CMY, HSV and YCbCr [19].

REFERENCES


