COLOUR IMAGE ENCRYPTION

B Tech Thesis

Submitted in partial fulfillment for the award of the Degree of Bachelor of Technology in Electronics and Communication Engineering

by

AVINASH S NAIR (Roll No.EC04B006)
JAYAKRISHNAN H (Roll No.EC04B036)
JITHIN R J (Roll No.EC04B081)

Under the guidance of

Mrs. DEEPTHI P P

Department of Electronics Engineering
NATIONAL INSTITUTE OF TECHNOLOGY CALICUT
NIT Campus P.O., Calicut - 673601, India
2007
This is to certify that the thesis entitled “COLOUR IMAGE ENCRYPTION” is a bona fide record of the major project done by AVINASH S NAIR (Roll No.EC04B006), JAYAKRISHNAN H (Roll No.EC04B036) and JITHIN R J (Roll No. EC04B081) under my supervision and guidance, in partial fulfillment of the requirements for the award of Degree of Bachelor of Technology in Electronics & Communication Engineering from National Institute of Technology Calicut for the year 2007.

Mrs DEEPTHI P P
(Guide)
Sr lecturer
Dept. of Electronics and Communication Engineering

Dr LILLYKUTTY JACOB
Professor & Head
Dept. of Electronics and Communication Engineering

Place: NIT Calicut
Date: 26-4-2008
INTRODUCTION

Cryptographic solutions should be used to ensure privacy and confidentiality when personal visual information is transmitted over untrusted or public networks. Due to their simplicity and adequate information security, secret sharing schemes are often utilized for image data encryption. The so-called K:N visual secret sharing (VSS) scheme encrypts the input image by splitting the original content into N noise-like shares which can be then distributed over untrusted communication channels. Secret information is recovered by visually inspecting the stacked shares without the need for complicated cryptographic operations.

This stacking technique was the earlier concept nowadays various mathematical operations are used to perform the same. The decryption is carried out by using a prescribed number \((K, K<N)\) of subset shares. Fewer than K shares are insufficient to reconstruct the original data. The scheme generates K shared key images from a given binary image as a printed page and a transparency of the same size. When the transparency is stacked on top of the printed page, the original image is formed. The two keys are generated in such a way that the share images are “random” looking with no semblance to the original image. It is a highly secure mechanism since the decryption is being performed by the human visual system when both shares are brought together.

Here firstly we have splitted the images into 2 shares by performing bit level operations in the spatial domain ie. directly onto the pixel values. Taking advantage of the vectorial nature of the color image representation, the scheme operates on decomposed binary vector fields of the multi-dimensional input and utilizes the complete RGB gamut to
generate the color shares. The vectorial nature of the color images and the bit-level based processing allows for the development of secret shares solutions which can be used to protect consumer-grade color images, digital documents containing color art works. These secret shares so obtained satisfy the perfect reconstruction property, a property which is not attained by the conventional VSS-based solutions, perform cryptographic operations at the decomposed bit levels, utilize blocks of share bits producing shares with increased spatial resolution compared to the image’s resolution. But an increased number of bits is needed for the transmission of the spatially expanded image shares, the current schemes could not be considered cost-effective encryption solutions for distributing visual information over untrusted networks. So a new color image secret sharing scheme is used here. This secret sharing scheme satisfies the perfect reconstruction property and performs binary cryptographic operations on color vectors at the decomposed bit levels. Since it encrypts each binary component of the decomposed original vectors with a binary output instead of the usual block of bits, the produced shares have the same spatial resolution as that of the original image. Thus, the encrypted visual information can be transmitted over untrusted channels at a reasonable cost. By modifying both the spatial and spectral characteristics of the vectorial input in the decomposed binary domain the encryption procedure generates random-like color vectors which differ significantly from the original color inputs in both magnitude and orientation. The input is recovered only if a decryption procedure is utilized at the decomposed bit levels.

Large number of imaging applications including digital photography, archiving, and internet communications primarily use images stored in the JPEG format. Application of the existing shared key cryptographic schemes for these images requires conversion back into spatial domain. When transmission and storage of the shares are concerned, one may not necessarily be able to apply lossy compression techniques since the loss may result in an inability to decrypt. Hence, encryption techniques applicable in the compressed
domain are needed. So we use a shared key algorithm that works directly in the JPEG domain, thus enabling shared key image encryption for a variety of applications. The scheme manipulates the quantized DCT coefficients and the resulting noise-like shares are also stored in the JPEG format. The decryption process that combines the shares is lossless and hence the original JPEG file’s fidelity is preserved. Our results indicate that each share image is approximately the same size as the original JPEG retaining the storage advantage provided by JPEG.

Then in this project we have also included extensions of image encryption. A quantized, coefficient with zero value can produce non-zero valued shares according to the random assignment scheme; by assigning the same values for both shares. Using this technique, it is possible to introduce new coefficients at various different frequency locations where there are zero coefficients in the original image. The only constraint we impose are that the shares have to have the exact same bit pattern for these inserted coefficients. Images with smooth regions with dominant chroma components and high contrast regions, the base scheme produces shares that show some envelope of the underlying picture. This is important since some applications may need unrecognizable shares.

Another extension is shares with unequal size. In this after taking the DCT the large coefficients are quantized and coded where as the small coefficients are coded as such and send.

Then we also implemented cheating prevension. Even with the remarkable advance of computer technology, using a computer to decrypt secrets is infeasible in some situations. For example, a security guard checks the badge of an employee or a secret agent recovers an urgent secret at some place where no electronic devices are available. In these situations the human visual system is one of the most convenient and reliable tools to do checking and secret recovery. Therefore, Naor and Shamir invented the visual cryptography in which a secret image (printed text, picture, etc.) is encrypted in a perfectly secure way such that the secret can be decoded directly by the human visual system. VC is a method of encrypting a secret image into shares such that stacking a
sufficient number of shares reveals the secret image. Shares are usually presented in transparencies. Each participant holds a transparency (share). Unlike conventional cryptographic methods, VC needs no complicated computation for recovering the secret. The act of decryption is to stack shares and view the image that appears on the stacked shares simply. A visual cryptography scheme is a visual secret sharing scheme such that stacking any or more shares reveals the secret image, but stacking fewer than shares reveals not any information about the secret image.

In these cases, all participants who hold shares are assumed to be semi-honest, that is, they will not present false or fake shares during the phase of recovering the secret image. Thus, the image shown on the stacking of shares is considered as the real secret image. Nevertheless, cryptography is supposed to guarantee security even under the attack of malicious adversaries who may deviate from the scheme in any way. We have seen that it is possible to cheat in VC, though it seems hard to imagine. For cheating, a cheater presents some fake shares such that the stacking of fake and genuine shares together reveals a fake image. With the property of unconditional security, VC is suitable for sending highly classified orders to a secret agent when computing devices may not be available. The secret agent arrived some shares, each with a pre-determined order, when departing to the hostile country. When the headquarter decides to execute a specific order, it can simply send another share to the agent so that the agent can recover what the order is. We can see that it would be terrible if the dispatched share cannot be verified due to a cheater’s attack.
SIGNIFICANCE

In a modern technological environment, the conventional VSS application scenario which requires the use of transparencies, an overhead projector and the human vision system properties to decrypt the secret [9,15,23] is not very appealing. Moreover, it does not consider the fact that the end-user requires the decrypted output to be available in a digital format for storage, transmission or further processing. Therefore, a different approach to image sharing is necessary. Combining the secret sharing concept presented in [9] with the bit-level decomposition and stacking operations from [28], a 2; 2g color image secret sharing scheme has been developed recently [22]. Taking advantage of the vectorial nature of the color image representation, the scheme operates on decomposed binary vector fields of the multi-dimensional input and utilizes the complete RGB gamut to generate the color shares. The vectorial nature of the color images and the bit-level based processing allows for the development of robust solution [10,16] which can be used to protect consumer-grade color images, digital documents containing color artworks, as well as visual data in the enterprise pipeline. All these solutions: (i) satisfy the perfect reconstruction property (the decrypted image is identical to the input image), a property which is not attained by the conventional VSS-based solutions, (ii) perform cryptographic operations at the decomposed bit levels, and (iii) utilize blocks of share bits producing shares with increased spatial resolution compared to the input (secret) image’s resolution. Since an increased number of bits is needed for the transmission of the spatially expanded image shares, the current schemes could not be considered cost-effective encryption solutions for distributing visual information over untrusted networks. To this end, a new color image secret sharing scheme is introduced in this work. The unique characteristic differentiates the proposed solution not only from
traditional VSS methods but from our previously published works. By modifying both the spatial and spectral characteristics of the vectorial input in the decomposed binary domain, the encryption procedure generates random-like color vectors which differ significantly from the original color inputs in both magnitude and orientation. The input is recovered only if a decryption procedure is utilized at the decomposed bit levels.

Large number of imaging applications including digital photography, archiving, and internet communications primarily use images stored in the JPEG format. Most of the digital cameras in the market use JPEG. Application of the existing shared key cryptographic schemes for these images requires conversion back into spatial domain. When transmission and storage of the shares are concerned, one may not necessarily be able to apply lossy compression techniques since the loss may result in an inability to decrypt. Hence, encryption techniques applicable in the compressed domain are needed. In this paper we propose a shared key algorithm that works directly in the JPEG domain, thus enabling shared key image encryption for a variety of applications. The scheme manipulates the quantized DCT coefficients and the resulting noise-like shares are also stored in the JPEG format. The decryption process that combines the shares is lossless and hence the original JPEG file’s fidelity is preserved. Our experiments indicate that each share image is approximately the same size as the original JPEG retaining the storage advantage provided by JPEG.
TECHNIQUE EMPLOYED

CODING
For the coding procedure the given colour image is split into small blocks of size 8x8 pixels. Each pixel value consists of a red, green and a blue component. These components are separated for each pixel. Each of this component is the red, green and blue for each pixel is binary encoded with 8 bits. Then each of this 8 bits are separated using the technique below using a random number generator. With this we get 2 shares which are then transmitted.

\[
\begin{bmatrix}
    b_i^{n1}(l) \\
    b_i^{n2}(l)
\end{bmatrix} \in \begin{cases}
    \{[0,1],[1,0]\} & \text{if } b_i^x(l) = 1 \\
    \{[0,0],[1,1]\} & \text{if } b_i^y(l) = 0
\end{cases}
\]

Block diagram of the above procedure is below.
DECODING

In the decoding procedure the 2 shares are xored. This will result in the recovery of correct symbol. Then all the 8 bits of the pixel value are collected and their corresponding decimal equivalent is found. This is done for all the three components i.e., red, green, and blue, and hence the corresponding pixel value is calculated.

\[
\begin{array}{c}
\begin{array}{c}
0 \text{ XOR } 0 \\
1 \text{ XOR } 1
\end{array}
\end{array}
\]

\[
\begin{array}{c}
\begin{array}{c}
0 \text{ XOR } 1 \\
1 \text{ XOR } 0
\end{array}
\end{array}
\]

Block diagram of the above procedure is below.
But there are certain problems associated with the above technique such as compression and transmission. Hence we go for another technique for encryption using JPEG. Advances over the past decade in many aspects of digital technology especially devices for image acquisition, data storage, and bitmapped printing and display have brought about many applications of digital imaging. However, these applications tend to be specialized due to their relatively high cost. With the possible exception of facsimile, digital images are not commonplace in general purpose computing systems the way text and geometric graphics are. The majority of modern business and consumer usage of photographs and other types of images takes place through more traditional analog means. The key obstacle for many applications is the vast amount of data required to represent a digital image directly. A digitized version of a single, color picture at TV resolution contains on the order of one million bytes; 35mm resolution requires ten times that
amount. Use of digital images often is not viable due to high storage or transmission costs, even when image capture and display devices are quite affordable. Modern image compression technology offers a possible solution. State of the art techniques can compress typical images from 1/10 to 1/50 their uncompressed size without visibly affecting image quality. But compression technology alone is not sufficient. For digital image applications involving storage or transmission to become widespread in today’s marketplace, a standard image compression method is needed to enable interoperability of equipment from different manufacturers. The CCITT recommendation for today’s ubiquitous Group 3 fax machines [17] is a dramatic example of how a standard compression method can enable an important image application. The Group 3 method, however, deals with bilevel images only and does not address photographic image compression. For the past few years, a standardization effort known by the acronym JPEG, for Joint Photographic Experts Group, has been working toward establishing the first international digital image compression standard for Continuous tone (multilevel) still images, both grayscale and color. The “joint” in JPEG refers to a collaboration between CCITT and ISO. JPEG convenes officially as the ISO committee designated JTC1/SC2/WG10, but operates in close informal collaboration with CCITT SGVIII. JPEG will be both an ISO Standard and a CCITT Recommendation. The text of both will be identical.

CODING
In this we use JPEG. The lossy version of JPEG image compression uses discrete cosine transforms. A monochrome image is first split into 8×8 non-overlapping blocks of pixels. An 8×8 DCT is applied to each block and the resulting coefficients are scalar quantized using a quantization matrix. The quantized coefficients are then converted from a two-
dimensional representation to a one-dimensional vector by a process known as zig-zag scanning and sent to an entropy coder that uses either Huffman or arithmetic coding. The process is shown below.

Comression using DCT uses following steps
- image is broken into 8x8 blocks of pixels
- working from left to right, top to bottom, the DCT is applied to each block.
- Each block is compressed through quantization.
- The array of compressed blocks that constitute the image is stored in a reduced amount of space.
- When desired the image is reconstructed through decomposition, a process that uses an inverse DCT.

The equation for DCT
In the matrix form the equation is

\[
D(i,j) = \frac{1}{\sqrt{2N}} C(i)C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} p(x,y) \cos\left[ \frac{(2x + 1)i\pi}{2N} \right] \cos\left[ \frac{(2y + 1)j\pi}{2N} \right]
\]

\[
C(u) = \begin{cases} 
\frac{1}{\sqrt{2}} & \text{if } u = 0 \\
1 & \text{if } u > 0 
\end{cases}
\]

In the matrix form the equation is

\[
D(i,j) = \frac{1}{4} C(i)C(j) \sum_{x=0}^{7} \sum_{y=0}^{7} p(x,y) \cos\left[ \frac{(2x + 1)i\pi}{16} \right] \cos\left[ \frac{(2y + 1)j\pi}{16} \right]
\]

To get the matrix form we use the following equations.

\[
T_{ij} = \begin{cases} 
\frac{1}{\sqrt{N}} & \text{if } i = 0 \\
\sqrt{\frac{2}{N}} \cos\left[ \frac{(2i+1)j\pi}{2N} \right] & \text{if } i > 0 
\end{cases}
\]

For a 8x8 block it results in this form.
DCT is accomplished by matrix multiplication.

\[ D = TMT' \]

**QUANTIZATION**

The next step is the quantization. 8x8 blocks of DCT are ready for quantization. Quantization is carried out by using Q matrix. Here we are using Q50. i.e., it can recover 50 percent of the details.

Q50 matrix is:

\[
T = \begin{bmatrix}
0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 \\
0.4904 & 0.4157 & 0.2778 & 0.0975 & -0.0975 & -0.2778 & -0.4157 & -0.4904 \\
0.4619 & 0.1913 & -0.1913 & -0.4619 & -0.4619 & -0.1913 & 0.1913 & 0.4619 \\
0.4157 & -0.0975 & -0.4904 & -0.2778 & 0.2778 & 0.4904 & 0.0975 & -0.4157 \\
0.3536 & -0.3536 & -0.3536 & 0.3536 & 0.3536 & -0.3536 & -0.3536 & 0.3536 \\
0.2778 & -0.4904 & 0.0975 & 0.4157 & -0.4157 & -0.0975 & 0.4904 & -0.2778 \\
0.1913 & -0.4619 & 0.4619 & -0.1913 & -0.1913 & 0.4619 & -0.4619 & 0.1913 \\
0.0975 & -0.2778 & 0.4157 & -0.4904 & 0.4904 & -0.4157 & 0.2778 & -0.0975
\]
Quantization is achieved by dividing each element in the transformed matrix $D$ by the corresponding elements in $Q$ matrix and then rounding out to the nearest integer value. For the following step, $Q_{50}$ is used.

$$Q_{50} = \begin{bmatrix}
16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\
12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\
14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\
14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\
18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\
24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\
49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\
72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 
\end{bmatrix}$$

The quantized $C$ is now ready for final step of compression. Before storage all the coefficients of $C$ are converted by an encoder to a bit stream of binary data (01101011...). In depth coverage of coding process is beyond the scope of this project. However, we can point out one key aspect that the reader is sure to appreciate. After quantization, it is quite common for most of the coefficients is equal to zero. JPEG takes advantage of this by encoding quantized coefficients in zig zag sequence as shown in figure below. The
advantage lies in the consolidation of relatively large runs of zeros, which compress very well.

\[
\begin{array}{c|c|c}
(0,0) & (0,1) & \\
(1,0) & (1,1) & \\
\end{array}
\]

Then each of this bits are separated using the technique below using a random number generator. With this we get 2 shares which are then transmitted.

\[
\begin{bmatrix}
b^n_1(l) \\
b^n_2(l)
\end{bmatrix} \in \begin{cases} 
[0,1], [1,0] & \text{if } b^n_1(l) = 1 \\
[0,0], [1,1] & \text{if } b^n_2(l) = 0
\end{cases}
\]
DECODING

In the decoding procedure the 2 shares are xored. From here we actually get back the bits in C matrix. Reconstruction of our image begins with decoding the bit stream representation of quantized matrix C. Each element of C is then multiplied by the corresponding element of the quantization matrix originally used.

\[ R_{i,j} = Q_{i,j} \times C_{i,j} \]
IDCT is next applied to R, which is rounded to the nearest integer. Finally 128 is added to each element of that result, giving us the decompressed image giving us decompressed JPEG versions of our original 8x8 image block M.

\[ N = \text{round}(T \cdot R \cdot T) + 128 \]

Decoding block diagram.

**EXTENSIONS**

The technique proposed above can be further extended for several improvements. In this section, we discuss such extensions. One is improvement of appearance of randomness in the shares when needed. The method as described so far is a \( \{2,2\} \) sharing scheme. We will point out how this can be extended to a \( \{n,k\} \) sharing scheme with relatively simple additional steps. We will also detail a method for generating asymmetric shares such that the size of one share is only a fraction of the other’s or that of the original. In the
following subsections, these extensions are described with corresponding experimental results. With these extensions, the application possibilities of the proposed method significantly improve.

Insertion of coefficients
If we needed the ability to change the appearance of the resulting shares, we should be able to control the quantized DCT coefficients and their distributions. To achieve that, we make an important observation. A quantized, DCT coefficient with zero value can produce non-zero valued shares by assigning the same values for both shares. Using this technique, it is possible to introduce new coefficients at various different frequency locations where there are zero coefficients in the original image. When introducing the coefficients, the only constraints we impose are that the shares have to have the exact same bit pattern for these inserted coefficients and that the inserted values do not exceed the 12bits (baseline JPEG restriction). For images with smooth regions with dominant chroma components and high contrast regions, the base scheme produces shares that show some envelope of the underlying picture. While this may be acceptable for many applications, some applications may demand unrecognizable noise-like structures for the shares. To arrive at noise-like shares, we use the insertion mechanism as described above. We insert new coefficients in the general region of spatial frequency points where the human visual sensitivity is high. For the luminance components, the sensitivity starts increasing at the DC location and peaks at around the first few coefficients of the zig-zag scanned AC coefficient. For chroma components, this region includes the DC and a small region surrounding. Given such sensitivity patterns, we introduce new nonzero coefficients in the shares where the original has zero values in the regions closer to the DC component. The amplitudes and signs of the introduced coefficients were chosen randomly. We selected the blocks that have their total energy below a threshold for candidates for insertion of coefficients.

\[ E = \sum_{i=0}^{63} |X_i^n| < \text{Threshold}, \]
The number of inserted coefficients is also chosen in an ad hoc manner using several experiments. An extension of this method may use information from the standard human visual studies as an aid to automatically determine which blocks need to be inserted with coefficients.

Asymmetric Shares

The base mechanism we discussed so far results in producing shares with file sizes roughly equal to the original file size. Some applications may prefer an asymmetry in the resulting file sizes for security or communication bandwidth reasons. The file sizes depend on the number of coefficients and their distribution in each DCT block. Again, the DC component and the AC coefficients near it contain the bulk of the information in a picture and by selecting a few of the lower frequency components for share generation, shares with asymmetric sizes can be obtained. Share 1 was generated considering only the DC and the first AC coefficient alone. Share 2 contains the other shares of these coefficients and the rest of the AC coefficients unmodified from the original image. Share 1 contains less information and does not reveal anything significant regarding the original image. Share 2 however retains the high frequency information and hence shows some contours of the image.

Cheating Prevention

Starting of with the basics all participants who hold shares are assumed to be semi-honest, that is, they will not present false or fake shares during the phase of recovering the secret image. Thus, the image shown on the stacking of shares is considered as the real secret image. Nevertheless, cryptography is supposed to guarantee security even under the attack of malicious adversaries who may deviate from the scheme in any way. We have seen that it is possible to cheat in VC, though it seems hard to imagine. For cheating, a cheater presents some fake shares such that the stacking of fake and genuine shares
together reveals a fake image. With the property of unconditional security, VC is suitable for sending highly classified orders to a secret agent when computing devices may not be available. The secret agent carried some shares, each with a pre-determined order, when departing to the hostile country. When the headquarter decides to execute a specific order, it can simply send another share to the agent so that the agent can recover what the order is. We can see that it would be terrible if the dispatched share cannot be verified due to a cheater’s attack.

Hence the technique we employed here is we actually XOR the shares we got with an authentication image and then send through the channel. At receiver its XORed again with the share and if we get back the authentication image then its safe to send the image and hence the shares are transmitted.

RANDOM NUMBER GENERATOR

The pseudorandom noise (PN) sequences are a series of 1’s and 0’s which lack any definite pattern, and look statistically independent and uniformly distributed. The sequences are deterministic, but exhibit noise properties similar to randomness. Gold codes are obtained by combining two PN sequences and modulo-2 adding, or XORing, the output together. These codes have good cross-correlation properties. The pseudorandom noise (PN) sequences are a series of 1’s and 0’s which lack any definite pattern, and look statistically independent and uniformly distributed. The sequences are
deterministic, but exhibit noise properties similar to randomness. The PN sequence generator is usually made up of shift registers with feedback. By linearly combining elements from taps of the shift register and feeding them back to the input of the generator, you can obtain a sequence of much longer repeat length using the same number of delay elements in the shift register. Hence, these blocks are also referred to as linear feedback shift registers (LFSR). The length of the shift register, the number of taps, and their positions in the LFSR, are important to generate PN sequences with desirable auto correlation and cross-correlation properties. The main principle of spread spectrum communication is using wide band, noise-like signals to increase the bandwidth occupancy. As a result of larger bandwidth, the power spectral density is lower, which enables multiple signals to occupy the same band with minimum interference. During the spreading process, CDMA distributes the signal across the entire allotted frequency spectrum by combining the data signal with a scrambling code which is independent of the transmitted signal. In a multi-path environment, each addressee is assigned a unique scrambling code. The correlation property of these codes makes it possible to generate a distinction between the signals, which allows the different paths to be decoded by the receiver.

The X sequence uses the following code

\[ X^{25} + X^3 + 1 \]

The Y sequence uses the following code

\[ X^{25} + X^3 + X^2 + X + 1 \]

Figure below shows the uplink long scrambling code generator block diagram
RESULTS

1. SPATIAL DOMAIN
3. INSERTION OF COEFFICIENTS

DCT IMAGE

SHARES

RECONSTRUCTED DCT IMAGE

RECONSTRUCTED IMAGE

4. ASSYMETRICAL SHARES
5. CHEATING PREVENTION
REFERENCES

- Rastislav Lukac_, Konstantinos N. Plataniotis “ A cost effective encryption scheme for colour images.
- Chih-Ming Hu and Wen-Guey Tzeng “Cheating Prevention in Visual Cryptography