Islamic University – Gaza
Deanery of Higher Studies
Faculty of Engineering
Computer Engineering Department

# Selective Encryption of Images Using Differential Evolution

**Nadeen Salem Deeb**

**Supervisor**

**Prof. Ibrahim S. I. Abuhaiba**

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Engineering

1432H (2011)

## Dedication

**To those whose kindness, patience, and support were the candles that enlightened my way towards success; my Father and Mother.**

**To my beloved husband who saved no efforts in encouraging and supporting me during my journey towards success, and to his extended family.**

**To my brothers and my sisters who spiritually supported me.**

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

## LIST OF ACRONYMS

| | |
|---|---|
| **AC coefficient** | All the reaming components in the block , exclude the dc component |
| **AES** | Advanced Encryption Standard |
| **CE** | Consumer Electronics Devices |
| **CF** | Compression Friendliness |
| **CS** | Cryptographic Security |
| **DC coefficient** | Mean value of the block |
| **DCT** | Discrete Cosine Transform |
| **DE** | Differential Evolution |
| **DES** | Data Encryption Standard |
| **EAs** | Evolutionary Algorithms |
| **ER** | Encryption Ratio |
| **FC** | Format Compliance |
| **FDCT** | Forward Discrete Cosine Transform |
| **IDCT** | Inverse Discrete Cosine Transform |
| **IDEA** | International Data Encryption Algorithm |
| **JPEG** | Joint Photographic Experts Group |
| **LFSRs** | Linear Feedback Shift Registers |
| **MPEG** | Moving Picture Experts Group |
| **MSB** | Most Significant Bit |
| **MSE** | Mean Squared Error |
| **PDAs** | Personal Digital Assistants |
| **PRNG** | Pseudo Random Number Generator |
| **PSNR** | Peak Signal-To-Noise Ratio |
| **qDCi** | Quantized DC Coefficient of block i |
| **QF** | Quality Factor For Compression |
| **RSA** | Rivest, Shamir and Adleman public key encryption |
| **SE** | Selective Encryption |
| **T** | Tunability |
| **VD** | Visual Degradation |
| **VLC** | Variable-Length Code |
| **VLI** | Variable-Length Integer |
| **WDCM** | Watermarking by DC Coefficients Modification |
| **YCbCr** | Y is the Luma component and $C_b$ And $C_r$ are the blue-difference and red-difference chroma components |

# تشفيرانتقائي للصور باستخدام التطور التفاضلي

نادين سالم ديب

## ملخص

يعتمد أمن البيانات متعددة الوسائط في الشبكات الرقمية الموزعة على التشفير، وهي العملية الحسابية التي تحول الرسالة المفهومة الى أخرى مبهمة المعالم والدلائل. وفي المجمل فإن معظم طرق التشفير القديمة وحتى الحديثة منها قد صممت لحماية أبسط أنواع البيانات المتعددة كالنصوص، وهي غير قادرة على أن تستخدم بشكل فعال للأنواع الأخرى من البيانات المتعددة التي تمتاز بكبر حجمها مثل الصور و الفيديو.

لذلك ظهر حديثا مفهوم التشفير الانتقائي كنمط جديد للتشفير والذي يهدف إلى تشفير جزء من المعلومة من أجل تقليل المتطلبات الحاسوبية التي تحتاجها الكميات الضخمة من البيانات المتعددة الوسائط المنقولة خلال الشبكات الموزعة.

ومن هذا المنطلق، جاءت هذه الدراسة لتقدم نموذج جديد للتشفير الانتقائي للصور اعتمادا على مفهوم عمليتي التغيير والتبادل المستخدمتين في التطور التفاضلي.

ففي هذا النموذج، يتم تشفير الصور بكفاءة في مجال الترددات من خلال توظيف عملية التبادل الانتقائي لمعاملات محول جيب التمام من نفس نطاقات التردد بين القطاعات المنتقاه. وتحدد القطاعات المنتقاه في عملية التبادل بناءاً على قيمة قوة الارتباط داخل القطاع نفسه. أما عملية التغيير فينحصر عملها على تغيير خانة العلامة ( التي تحدد ايجابية أو سلبية المعامل) لبعض معاملات محول جيب التمام لعدد معين من نطاقات التردد والقطاعات تحدد بناء على احتياجات المستخدمين. وأخيرا، تتم عملية تحجيم معاملات جيب التمام من أجل تفريق وتفكيك الإحصائيات الموجودة بين المعاملات.

وتنتج الأرقام العشوائية المستخدمة للتحكم في عملية التشفير من مولدات مؤمنة ومشفرة بحيث تساوي قيمتها الابتدائية جزءاً من المفتاح السري والذي يصل حجمه إلى ثمانين خانة.

وأثبتت الاختبارات التجريبية الشاملة التي أجريت على هذا النموذج الجديد أنه يتلاءم مع مجال واسع من التطبيقات التي تحتاج إلى مستويات مختلفة من الأمن والحماية والتعددية في الرؤية البصرية للمعلومات المشفرة. كما أنه يمتاز بمحدودية تأثيره السلبي على كفاءة الضغط، ويدعم التدرج وتنفيذ بعض دوال المحتويات دون الحاجة إلى امتلاك مفتاح التشفير لتنفيذ فك التشفير، بسبب توافقه مع نمط الضغط المستخدم. وبناءا على هذه المزايا، يعتبر هذا النموذج الجديد مناسباً جدا لنقل الصور عبر الشبكات الموزعة.

# Selective Encryption of Images Using Differential Evolution

**Nadeen Salem Deeb**

## ABSTRACT

The security of multimedia data in digital distribution networks is commonly provided by encryption, i.e., the mathematical process that transforms a plaintext message into unintelligible ciphertext. Nevertheless, the classical and modern ciphers have all been developed for the simplest form of multimedia data, i.e., text, and are not appropriate for higher forms such as images and video with very large file sizes. Selective encryption is a recent approach to reduce the computational requirements for huge volumes of multimedia data in distribution networks with different client device capabilities. In this research, we provide a new scheme of selective encryption of images based on Differential Evolution operations (crossover and mutation).

In this research, image data are encrypted efficiently in the frequency domain by employing selective crossover of DCT coefficients of the same frequency bands between selective segments. Correlation of the segment is used as a fitness function of the crossover to determine the selective segments. The mutation function modifies the sign bit of certain DCT coefficients restricted to a specific number of blocks and segments determined by the user needs. Finally, Scaling DC coefficient is done to diffuse statistics. Cryptographically, secure pseudo-random number generator whose seed values are the sub-keys generated from a 80-bits secret key is used to control the entire encryption process.

Thorough experimental tests are carried out with detailed analysis demonstrating considerable levels of security and different levels of visual degradation to target different applications requirements. In additions, our new approach has very limited adverse impact on the compression efficiency and it allows scalability, and some other content processing functionalities without having to access the cryptographic key and to perform decryption, since it achieves format compliance. These advantages make it suitable for image transmission over network.

**Key words:** Selective Encryption, Differential Evolution, Crossover, Mutation, DCT Coefficients, Visual Degradation, Compression Efficiency, Format Compliance.

# CHAPTER 1: INTRODUCTION

## 1.1 INTODUCTION

The widening Internet bandwidth and availability of digital consumer electronics (CE) devices for playback, recording and storage have increased the demand for multimedia services. In the digital domain, distribution networks need to address two fundamental problems: (1) Reduction of huge communication requirements for multimedia data, and (2) Protection of copyrighted multimedia data. A solution to the first problem is provided by efficient coding techniques for images, audio and video. Compression [JPEG, JPEG 2000, MPEG-1, MPEG-2, MPEG-4, H.26X] removes the spatial and temporal redundancy in multimedia data with imperceptible degradation [1].

The second problem is addressed in privately defined closed systems by controlling access to copyrighted content. The CE devices that receive satellite and cable transmissions are equipped with the software and hardware needed to prevent unauthorized access.

Nowadays, images are widely used in several processes. Furthermore, special and reliable security in storage and transmission of digital images is needed in many applications, such as pay-TV, medical imaging systems, military image communications, confidential video conferences, etc. In order to fulfill such tasks, many image encryption methods have been proposed. Image encryption plays a significant role in the field of information hiding. Image hiding or encrypting methods and algorithms range from simple spatial domain methods to more complicated and reliable frequency domain ones.

Image encryption techniques try to convert an image to another one that is hard to understand [2]. On the other hand, image decryption retrieves the original image from the encrypted one. There are various image encryption systems to encrypt and decrypt data, and there is no single encryption algorithm that satisfies the different image types.

Some recent works explored a new way of securing the content, named, partial encryption or selective encryption by applying encryption to a subset of a bitstream. The main goal of selective encryption is to reduce the amount of data to encrypt while achieving a required

level of security. An additional feature of selective encryption is to preserve some functionalities of the original bitstream (e.g., scalability). The general approach is to separate the content into two parts [3].

Many algorithms for selective encryption have been proposed but they usually require a proprietary decoder which is unsuitable in the field of image transmission where JPEG standards dominate the market, and hence affect negatively on realtime applications.

## 1.2 RESEARCH MOTIVATIONS

Although an important and rich variety of selective encryption algorithms have been proposed in the literature, most of them suffer one or more of the following problems that may have a serious negative impact on a given application [4]:

1. Insufficient security

2. Decrease in the compression performance

3. Insignificant computational reduction with respect to total encryption

4. Lack of bitstream compliance

5. Increase in key size

6. Static definition of encrypted part

Hence, we are motivated to design the new selective encryption algorithms that can overcome most of the previous drawbacks which negatively impact on realtime applications.

## 1.3 GOAL, SCOPE, AND OBJECTIVES OF THE RESEARCH

There is a wide spectrum of secure multimedia applications with different requirements. They range from military applications that mandate total data obscurity to applications where a part of the multimedia data needs to be visible to allow searching in a shared

database. Hence, the goal of this research is to design, develop and evaluate a new approach of selective encryption of images using Differential Evolution approach (crossover and mutation). The research scope focuses on JPEG compressed and uncompressed images of gray scale and color space.

The desirable objectives of such research include:

- It should provide sufficient security for a range of multimedia applications.
- It should preserve the size of the original unencrypted bitstream.
- It should produce a bitstream that is compliant to the standard formats.
- It should not create a key whose size is much longer than those of commonly used modern ciphers.
- It should dynamically define the encrypted part with respect to image property and application requirements.
- It should identify the portions of the multimedia data to be encrypted.
- It should minimize the size of the encrypted part with respect to the whole data size.

In summary, this research aims to design, develop and evaluate a new approach of the selective encryption of JPEG compressed and uncompressed images using Differential Evolution approach [5, 6]. The proposed framework has the nice feature that the encryption process (mutation and crossover) is very simple and efficient. It provides different levels of security. It has very limited adverse impact on the compression

efficiency and on the size of the original unencrypted bitstream. Our method will be compliant to JPEG standard format, and defines the encrypted part based on the image property and required security.

## 1.4 RESEARCH STRUCTURE

This thesis is organized as follows:

Chapter 2 is intended to give an overview of JPEG compression standard. It also provides the basic concepts of Differential Evolution steps. In addition, the selective encryption with its potential applications is presented in this chapter.

Chapter 3 describes the architecture of our proposed scheme. Also, it provides the procedure of key generation used in the proposed scheme. The flowchart of the proposed selective encryption/decryption algorithm is presented for compressed and uncompressed cases.

Chapter 4 discusses the security analysis of the proposed image encryption scheme including some important tests like visual testing, sensitivity analysis, and key space analysis, etc., to prove that the proposed cryptosystem is secure against the most common attacks.

In Chapter 5, the comparison between our scheme and other related works is introduced based on specific evolution criteria such as tunability, visual degradation, encryption ratio, cryptographic security, compression friendliness, and format compliance.

Finally, Chapter 6 presents the conclusion of this research. Also, it gives suggestions for possible future work based on the proposed model.

# CHAPTER 2: OVERVIEW AND LITERATURE REVIEW

In this chapter, an overview of digital images, JPEG compression standard, Differential Evolution operations (crossover and mutation), and selective encryption are given to make the reader familiar with the research study area.

## 2.1 Digital Images

A digital image is defined as an array of individual pixels and each pixel has its own value. The array, and thus the set of pixels, is called a bitmap. If we have an image of 512 pixels × 512 pixels, it means that the data for the image must contain information about 262144 pixels [7].

Digital images are produced through a process of two steps: *sampling* and *quantization*. Sampling is the process of dividing the original image into small regions called pixels, whereas quantization is the process of assigning an integer value (i.e. color) to each pixel [7].

The number of colors (i.e. color space) that can be assigned to any picture element or pixel is a function of the number of bits, which is sometimes referred to as the color depth or bits resolution. This concept is also known as bits per pixel (bpp) and represents the color for each value. The color space is computed using the following equation

$$\text{ColorSpace} = 2^{b} \tag{2.1}$$

where: b is the bit depth.

The color values used in each bitmap depend on the specific bitmap format. This means that each pixel in a bitmap contains certain information, usually interpreted as color information. The information content is always the same for all the pixels in a particular bitmap. Thus, each color value in a bitmap is a binary number. A binary number is a series of binary digits that can be either 0 or 1 and called bits. This binary number in a given format will differ in length depending on the color depth of the bitmap, where the color depth of a bitmap determines the range of possible color values that can be used in each pixel. For example, each pixel in a 24-bit image can be one of roughly 16.8 million

colors. This means that each pixel in a bitmap has three color values between 0 and 255 and then those colors are formed by mixing together varying quantities of three primary colors: red, green and blue [7].

As the number of bits increases, the image quality is also increased. However, storage requirements will increase, resulting in a direct relationship between the image storage size and the bits resolution. Image storage size for an uncompressed image is computed using the following equation:

$$IMGSS = IMGR \times BR \tag{2.2}$$

where:

*IMGSS*: Image storage size

*IMGR*: Image resolution (i.e. image width × image height)

*BR*: Bits resolution (bits depth)

For example, the storage size of a 640 pixels × 480 pixels, true colored image is given as follows:

$IMGSS = W \times H \times BR = 640 \times 480 \times 24$ bits = (7372800/1024/8) = 900 *KB*.

## 2.2 JPEG COMPRESSION STANDARD

### 2.2.1 JPEG Overview

As digital technology has advanced at such a rapid pace, storage and communication of digital material, in particular digital images poses a problem. In order to address this, compression is used. There are two main categories of compression, namely lossy and lossless compression.

Lossless compression algorithms, e.g. IF. BMP and RLE, reduce the size of the file by relying on elimination of the redundancy in the file. It is also known as the noiseless

coding, entropy coding, and data compaction codes and these algorithms can perfectly recover original image.

Lossy compression implies that the reconstructed image will look almost identical to the original image but is not identical and allows for a significant reduction in size. Most images contain a lot of redundant information to the human eye, which is sensitive to some types of distortion and less sensitive to other types of distortion in an image. Good compression algorithms comprehend how human perception works and this should be taken into consideration in the design of a lossy image compression algorithm.

JPEG, adopted by the Joint Photographic Experts Group, presents a simple lossy technique referred to as the Baseline method. Algorithms such as JPEG discard much of the original information from an image such that you cannot reconstruct the exact original image from a JPEG file [8]. This is information that is not really noticed by human users and can thus be discarded for compression purposes. JPEG is mainly used on continuous tone images such as photographs. Since this kind of images do not contain much redundancy, pure lossless compression does not achieve significant reductions in size [9].

The JPEG standard has become a popular format for images on the Internet since digital images produced by digital cameras are often stored in the JPEG format. JPEG can obtain higher compression ratios on images with high pixel resolutions, simply because the degradation doesn't as noticeable on a big image. In the case of JPEG, the loss of specific picture detail is not detectable to the untrained human eye until compression ratios reach a high of 80-90 percentage ranges [9, 10]. The noticeable distortion is normally called an artifact. The following section details the aspects of JPEG compression.

### 2.2.2 The JPEG Image-Coding Standard

The JPEG standard is a widely used standard for lossy compression of still images [10, 11]. The JPEG standard can be applied to both grayscale and color images and achieves very good to excellent image quality. This standard can be implemented in software with an acceptable computational complexity.

JPEG first converts the image to be compressed into the YCbCr color space and each color plane is broken into 8x8 pixel blocks [12, 13]. The transformation into the YCbCr will not be explored as this dissertation will focus on single component (grayscale) image compression, because the grayscale plane includes more important characteristics of the image than the colored plane.

The blocks are then DCT (Discrete Cosine Transform) transformed and the DCT coefficients are divided by some predefined quantization values and rounded to the nearest integer according to a quality factor, the quantization values can be scaled by a constant [10]. The resulting quantized DCT coefficients are compressed by means of an entropy encoder such as Huffman or arithmetic coding. An inverse DCT is then performed to reconstruct the data resulting in an image very similar to the original one. If the quantization values were set properly, the human eye should perceive no visible difference.

The JPEG encoder consists of three main blocks as depicted in Figure 2.1.

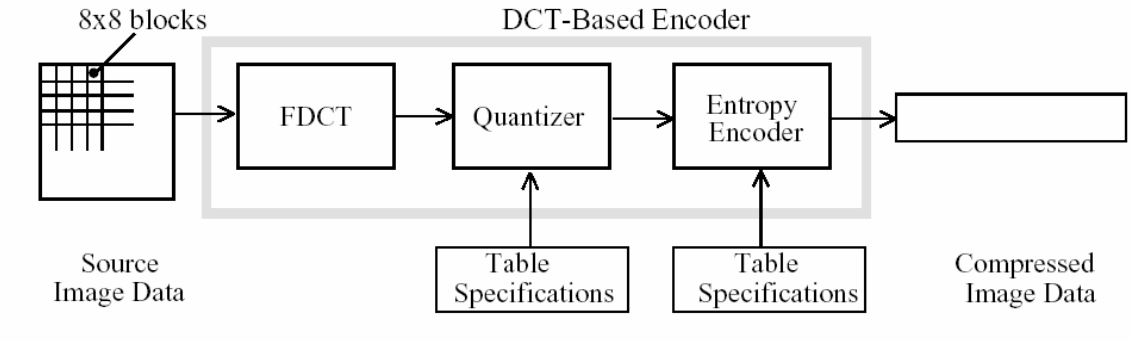



Figure 2.1 JPEG: DCT–based encoder processing steps [10]

The Decoder process is the inverse of the encoder process and is depicted in Figure 2.2 below. This uses the compressed image data to reconstruct the image using the inverse processes such as the Inverse Discrete Cosine Transform (IDCT).

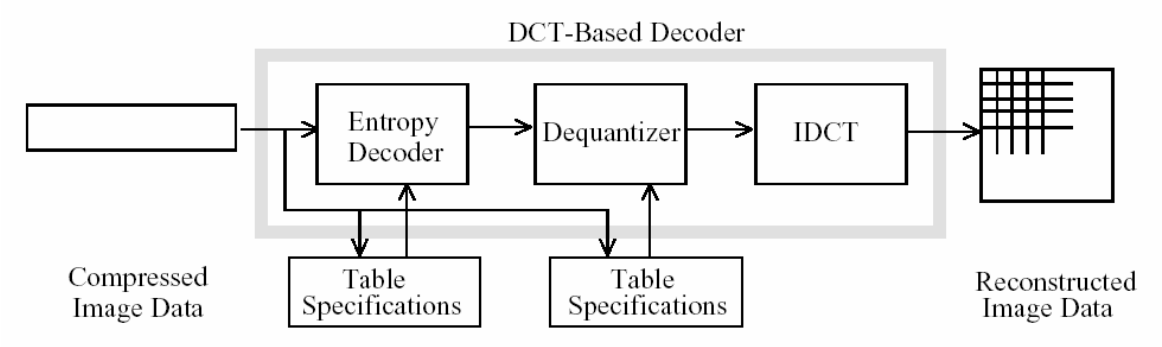



Figure 2.2 JPEG: DCT–based decoder processing steps [10]

The following sections explore each processing step in detail and how each step contributes to the final outcome of the JPEG image-coding standard [10, 8].

### 2.2.3 The DCT Component

JPEG makes use of transform coding techniques. Transform coding techniques compress the transform of a signal i.e. an image, and not the image directly. The Discrete Cosine Transform (DCT) is the most commonly used transform technique for image coding [14]. The energy compaction property of the DCT results in transform coefficients with only a few of the coefficients having significant values, thus making it a popular technique.

Each color component of a continuous tone image can be represented as a series of amplitudes in the two dimensional space. The DCT is used to discard higher frequency information that has minimal visual effect on the image. The DCT is related to the Discrete Fourier Transform (DFT) but can approximate linear signals well with few coefficients.

An image consists of many pixels arranged in an m x n array. The first step in the Forward Discrete Cosine Transform (FDCT) transformation of the image is to divide the picture array into 8x8 blocks of pixels. The size of the blocks has been chosen as a compromise of complexity and quality. If the number of rows or columns is not a multiple of 8, the closest multiple of 8 rows or columns is considered when dividing the image into

8x8 blocks. A comparison of the color data but not the intensity level of each pixel to its neighbors is performed [10]. The DCT produces large discrete coefficients for a pixel if the differences between the pixels are large. Otherwise, the differences are little, small DC coefficients are produced. The large amplitudes signify a large color difference. The pixels have thus been mapped from the spatial domain to the frequency domain.

This is in essence where the fundamental principle of JPEG lies, in eliminating the subtle color changes that the human eye cannot detect. In the frequency domain, this map to the smaller coefficients and thus these coefficients can be eliminated or rounded to zero. The data which contains the significant information is then retained by keeping the medium and larger coefficients [10].

Before computing the DCT of the 8×8 block, its values are shifted from a positive range to one centered around zero. For an 8-bit image, each entry in the original block falls in the range [0,255]. The mid-point of the range (in this case, the value 128) is subtracted from each entry to produce a data range that is centered around zero, so that the modified range is [-128,127]. This step reduces the dynamic range requirements in the DCT processing stage that follows.

In order to calculate the two-dimensional DCT coefficients, the general equation is given as

$$F(u,v) = (2/N)[C(u)C(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y) * \cos\frac{(2x+1)u\pi}{2N}\frac{\cos(2y+1)v\pi}{2N}] \quad (2.3)$$

Where N is the number of blocks of pixels according to which the image will be divided. $C(u),\ C(v) = 1/\sqrt{2}$ , when u = 0 and v = 0 [10], and $C(u),\ C(v) = 1,$ otherwise, f(x, y) represents the values of the 8x8 subarray block whose DCT is being calculated.

In computing the DCT of each subarray, 64 DCT coefficients are generated. F (0, 0) is known as the DC component and the remaining components are referred to as AC components, as shown in the Figure 2.3.

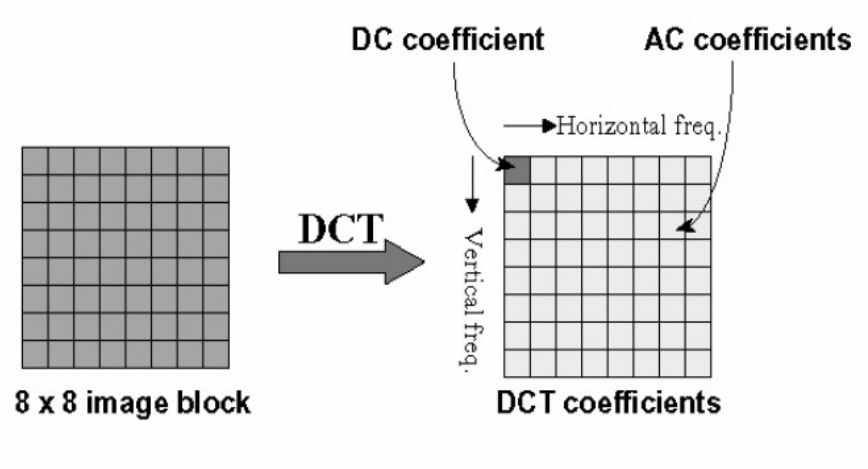



Figure 2.3 DCT coefficients of 8x8 image block [14]

At the decoder the IDCT reverses this processing step by reconstructing a 64 point output image signal by summing the basis signal [14]. In order to perform the reverse mapping from the frequency domain to the spatial domain, the following equation is used:

$$f(u,v) = (2/N)[C(u)C(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}F(x,y) * \cos\frac{(2x+1)u\pi}{2N}\cos\frac{(2y+1)v\pi}{2N}] \quad (2.4)$$

This DCT processing step is then followed by the quantization process which is explored below.

### 2.2.4 Quantization

The completion of a DCT results in the low frequency components of the coefficients having significant values. The DC component contains a significant amount of energy and there is a correlation between DC components of the current and preceding subarrays. A Uniform differential quantization scheme is used for the quantization of DC components and AC components are quantized using uniform quantization schemes [10, 14]. This is detailed later in this section.

The purpose of quantization is to modulate the influence of the different spectral components on the image. The influence of the highest DCT coefficients, which normally contain noise, is reduced. In essence, each block is quantized using a quantization table

meaning that each DCT coefficient is divided by its corresponding quantizer step size (an integer between 1 and 255) and rounded to the nearest integer. This is mathematically computed using the following equation [8].

$$F^{Q}(u,v) = \text{Integer Round}\left(\frac{F(u,v)}{Q(u,v)}\right) \tag{2.5}$$

Where Q (u, v) is the quantizer step size as per the quantization table, F (u, v) is the DCT coefficient, and $F^{Q}(u,v)$ is the resulting quantized coefficient.

JPEG compresses the color information, or "chrominance", in an image separately from the actual details of shapes, or "luminance". Luminance amounts to a grayscale image, while the chrominance amounts to a wash of colors painted on top of that grayscale image. The eye is much more sensitive to the details of shapes than color information, so the chrominance information can be compressed to a greater level than the luminance information. This means that JPEG will usually have higher compression ratios for color images than for grayscale images [8].

An example of a luminance quantization table is depicted in Figure 2.4.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 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 |

Figure 2.4 Quantization table [10]

The quantization step discards the information which is not visually significant and is thus the key source of lossiness in DCT-based encoders. The selection of the quantization table presents a trade-off between the compression achieved and the quality of the image. Dequantization is the inverse function which involves multiplying by the quautizer step

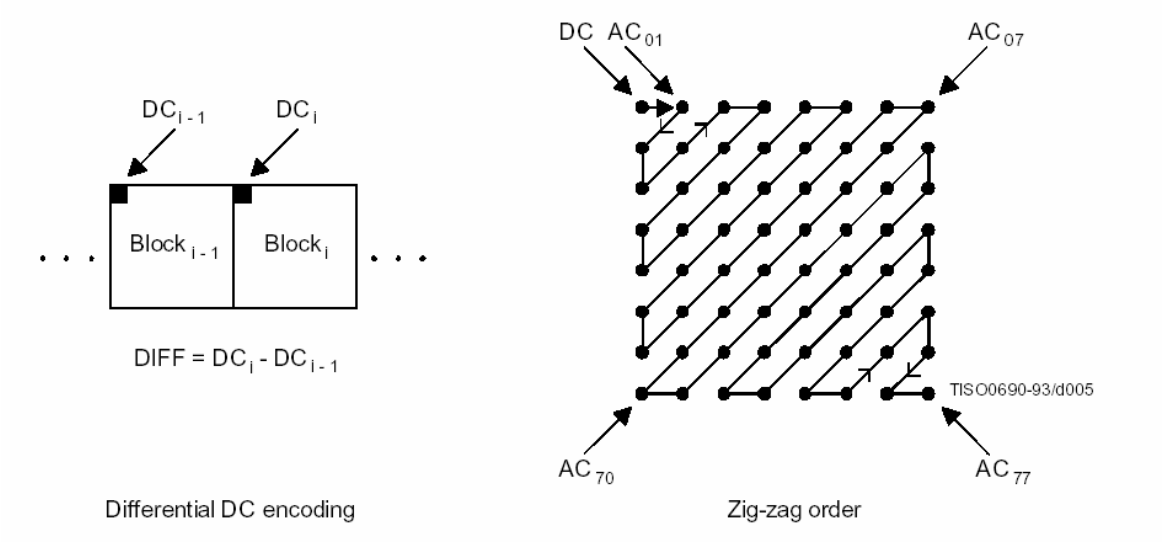
size as indicated in Equation 2.6. This result in a form appropriate as input to the IDCT. The quantization table is stored in the file header for use by the decompressor [10].

$$F(u,v) = F^{Q}(u,v) * Q(u,v) \tag{2.6}$$

Quantization thus enables lossy compression of an image by representing DCT coefficients with no greater precision than what is essential to achieve the required image quality.

### 2.2.5 DC Coding And Zig Zag Sequence

Once the quantization process has been performed, the DC coefficient is encoded as difference from the DC term of the previous block in the encoding order as depicted in Figure 2.5(a) since there is a strong correlation between the DC coefficients of adjacent 8x8 blocks [10].




(a) (b)

Figure 2.5 Preparation of quantized coefficients for entropy coding: a) Differential DC encoding, and b) Zigzag sequence [10]

The quantized DCT coefficients are arranged in a vector by zigzag sampling as shown in Figure 2.5 (b). This sampling results in a vector of length 64 with non-zero values populating only the first few components. This enables to facilitate entropy

encoding by placing low frequency coefficients before high frequency coefficients [15].

### 2.2.6 The Entropy Coding

The final DCT-based encoder processing step is entropy coding. This step achieves additional compression losslessly by encoding the quantized DCT coefficients more compactly based on their statistical characteristics. In the design, we will use the baseline sequential codec, which uses Huffman coding [16].

It is useful to consider entropy coding as two steps process. The first step converts the zig-zag sequence of quantized coefficients into an intermediate sequence of symbols. The second step converts the symbols to a data stream in which the symbols no longer have externally identifiable boundaries. The form and definition of the intermediate symbols are dependent on both the DCT-based mode of operation and the entropy coding method.

Huffman coding requires that one or more sets of Huffman code tables be specified by the application. The same tables used to compress an image are needed to decompress it. Huffman tables will be predefined and used within an application as defaults.

The entropy coding can be divided into the following steps:

#### 2.2.6.1 Intermediate Entropy Coding Representations

In the intermediate symbol sequence, each nonzero AC coefficient is represented in combination with the ‘‘runlength’’ (consecutive number) of zero-valued AC coefficients which precede it in the zig-zag sequence. Each such runlength/nonzero-coefficient combination is (usually) represented by a pair of symbols [8, 10]:

| Symbol-1 | Symbol-2 |
|---|---|
| (RUNLENGTH, SIZE) | (AMPLITUDE) |

Symbol-1 represents two pieces of information, RUNLENGTH and SIZE. Symbol-2 represents the single piece of information designated AMPLITUDE, which is simply the amplitude of the nonzero AC coefficient. SIZE is the number of bits used to encode AMPLITUDE that is, to encoded symbol-2, by the signed-integer encoding used with JPEG's particular method of Huffman coding [10].

RUNLENGTH represents zero-runs of length 0 to 15. Actual zero-runs in the zig-zag sequence can be greater than 15, so the symbol-1 value (15, 0) is interpreted as the extension symbol with runlength = 16. There can be up to three consecutive (15, 0) extensions before the terminating symbol-1 whose RUNLENGTH value completes the actual runlength. The terminating symbol-1 is always followed by a single symbol-2, except for the case in which the last run of zeros includes the last ($63^{rd}$) AC coefficient. In this frequent case, the special symbol-1 value (0, 0) means EOB (end of block), and can be viewed as an ''escape'' symbol which terminates the 8x8 sample block.

Thus, for each 8x8 block of samples, the zig-zag sequence of 63 quantized AC coefficients is represented as a sequence of symbol-1, symbol-2 symbol pairs, though each ''pair'' can have repetitions of symbol-1 in the case of a long run-length or only one symbol-1 in the case of an EOB.

Baseline sequential mode of JPEG encoding has 8-bit integer source samples in the range $[-2^7, 2^7-1]$, so quantized AC coefficient amplitudes are covered by integers in the range $[-2^{10}, 2^{10}-1]$ [10]. The signed-integer encoding uses symbol-2 AMPLITUDE codes of 1 to 10 bits in length (so SIZE also represents values from 1 to 10), and RUNLENGTH represents values from 0 to 15. For AC coefficients, the structure of the symbol-1 and symbol-2 intermediate representations is illustrated in Tables 2.1 and 2.2, respectively.

Table 2.1 Baseline Huffman coding symbol-1 structure [10]

| | SIZE | | | | | |
|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | . . . | 9 | 10 |
| RUN LENGTH | 0 EOB<br>. X<br>. X<br>. X<br>15 ZRL | | RUN-SIZE values | | | |

Table 2.2 Baseline entropy coding symbol2 structure [10]

| SIZE | AMPLITUDE |
|---|---|
| 1 | -1,1 |
| 2 | -3,-2,2,3 |
| 3 | -7..-4,4..7 |
| 4 | -15..-8,8..15 |
| 5 | -31..-16,16..31 |
| 6 | -63..-32,32..63 |
| 7 | -127..-64,64..127 |
| 8 | -255..-128,128..255 |
| 9 | -511..-256,256..511 |
| 10 | -1023..-512,512..1023 |

The intermediate representation for an 8x8 sample block's differential DC coefficient is structured similarly. Symbol-1, however, represents only SIZE information; symbol-2 represents AMPLITUDE information as before. Because the DC coefficient is differentially encoded, it is covered by twice as many integer values, $[-2^{11}, 2^{11-1}]$ as the AC coefficients, so one additional level must be added to the bottom of Table 2.2 for DC coefficients. Thus, symbol-1 for DC coefficients represents a value from 1 to 11 [8, 10].

#### 2.2.6.2 Variable-Length Entropy Coding

Once the quantized coefficient data for an 8x8 block is represented in the intermediate symbol sequence described above, variable-length codes are assigned. For each 8x8 block, the DC coefficient's symbol-1 and symbol-2 representation is coded and output first.

For both DC and AC coefficients, each symbol-1 is encoded with a variable-length code (VLC) from the Huffman table set assigned to the 8x8 block's image component. Each symbol-2 is encoded with a "variable-length integer" (VLI) code whose length in bits is given in Table 2.2. VLCs and VLIs both are codes with variable lengths, but VLIs are not Huffman codes. An important distinction is that the length of a VLC Huffman code is not known until it is decoded, but the length of a VLI is stored in its preceding VLC.

Huffman codes (VLCs) must be specified externally as an input to JPEG encoders. Note that the form in which Huffman tables are represented in the data stream is an indirect specification with which the decoder must construct the tables themselves prior to decompression.

### 2.2.6.3 Baseline Encoding Example

This section gives an example of Baseline compression [8] and encoding of a single 8x8 sample block.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 139 | 144 | 149 | 153 | 155 | 155 | 155 | 155 |
| 144 | 151 | 153 | 156 | 159 | 156 | 156 | 156 |
| 150 | 155 | 160 | 163 | 158 | 156 | 156 | 156 |
| 159 | 161 | 162 | 160 | 160 | 159 | 159 | 159 |
| 159 | 160 | 161 | 162 | 162 | 155 | 155 | 155 |
| 161 | 161 | 161 | 161 | 160 | 157 | 157 | 157 |
| 162 | 162 | 161 | 163 | 162 | 157 | 157 | 157 |
| 162 | 162 | 161 | 161 | 163 | 158 | 158 | 158 |

(a) source image samples

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 235.6 | -1.0 | -12.1 | -5.2 | 2.1 | -1.7 | -2.7 | 1.3 |
| -22.6 | -17.5 | -6.2 | -3.2 | -2.9 | -0.1 | 0.4 | -1.2 |
| -10.9 | -9.3 | -1.6 | 1.5 | 0.2 | -0.9 | -0.6 | -0.1 |
| -7.1 | -1.9 | 0.2 | 1.5 | 0.9 | -0.1 | 0.0 | 0.3 |
| -0.6 | -0.8 | 1.5 | 1.6 | -0.1 | -0.7 | 0.6 | 1.3 |
| 1.8 | -0.2 | 1.6 | -0.3 | -0.8 | 1.5 | 1.0 | -1.0 |
| -1.3 | -0.4 | -0.3 | -1.5 | -0.5 | 1.7 | 1.1 | -0.8 |
| -2.6 | 1.6 | -3.8 | -1.8 | 1.9 | 1.2 | -0.6 | -0.4 |

(b) forward DCT coefficients

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 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 |

(c) quantization table

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 15 | 0 | -1 | 0 | 0 | 0 | 0 | 0 |
| -2 | -1 | 0 | 0 | 0 | 0 | 0 | 0 |
| -1 | -1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

(d) normalized quantized coefficients

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 240 | 0 | -10 | 0 | 0 | 0 | 0 | 0 |
| -24 | -12 | 0 | 0 | 0 | 0 | 0 | 0 |
| -14 | -13 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

(e) denormalized quantized coefficients

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 144 | 146 | 149 | 152 | 154 | 156 | 156 | 156 |
| 148 | 150 | 152 | 154 | 156 | 156 | 156 | 156 |
| 155 | 156 | 157 | 158 | 158 | 157 | 156 | 155 |
| 160 | 161 | 161 | 162 | 161 | 159 | 157 | 155 |
| 163 | 163 | 164 | 163 | 162 | 160 | 158 | 156 |
| 163 | 164 | 164 | 164 | 162 | 160 | 158 | 157 |
| 160 | 161 | 162 | 162 | 162 | 161 | 159 | 158 |
| 158 | 159 | 161 | 161 | 162 | 161 | 159 | 158 |

(f) reconstructed image samples

Figure 2.6 DCT and Quantization example [8]

Figure 2.6(a) is an 8x8 block of 8-bit samples, arbitrarily extracted from a real image. The 8x8 block is input to the DCT, Equation 2.3. Figure 2.6(b) shows (to one decimal place) the resulting DCT coefficients. Except for a few of the lowest frequency coefficients, the amplitudes are quite small.

Figure 2.6(c) is the example quantization table for luminance (grayscale) components included in the informational annex of the draft JPEG standard. Figure 2.6(d) shows the

quantized DCT coefficients, normalized by their quantization table entries. At the decoder these numbers are “denormalized” according to Equation 2.6, and input to the IDCT, Equation 2.4.

Finally, Figure 2.6(f) shows the reconstructed sample values, remarkably similar to the originals in Figure 2.6(a). Of course, the numbers in Figure 2.6(d) must be Huffman-encoded before transmission to the decoder.

The first number of the block to be encoded is the DC term, which must be differentially encoded. If the quantized DC term of the previous block is, for example 12, then the difference are +3. Thus, the intermediate representation is (2) (3), for SIZE = 2 and AMPLITUDE = 3.

Next, the quantized AC coefficients are encoded. Following the zig-zag order, the first non-zero coefficient is -2, preceded by a zero-run of 1. This yields an intermediate representation of (1, 2) (-2). Next encountered in the zig-zag order is three consecutive non-zeros of amplitude -1. This means consecutive non-zeros of amplitude -1. This means each is preceded by a zero-run of length zero, for intermediate symbols (0, 1) (-1). The last non-zero coefficient is -1 preceded by two zeros, for (2, 1) (-1).

Because this is the last non-zero coefficient, the final symbol representing this 8x8 block is EOB, or (0, 0). Thus, the intermediate sequence of symbols for this example 8x8 blocks is:

(2)(3), (1, 2) (-2), (0, 1) (-1), (0, 1) (-1), (0, 1) (-1), (2, 1) (-1), (0, 0)

Next the codes themselves must be assigned. The differential-DC VLC for this example is:

(2) 011

The AC luminance VLCs for this example are:

(0, 0) 1010
(0, 1) 00

(1, 2) 11011
(2, 1) 11100

The VLIs are related to the two's complement representation. They are:

(3) 11
(-2) 01
(-1) 0

Thus, the bit-stream for this 8x8 example block is as follows. Note that 31 bits are required to represent 64 coefficients, which achieves compression of just under 0.5 bits/sample:

01111110110100000000001110001010

## 2.3 DIFFERENTIAL EVOLUTION

Differential Evolution (DE) algorithm is a new heuristic approach mainly having three advantages; finding the true global minimum regardless of the initial parameter values, fast convergence, and using few control parameters [6]. DE algorithm is a population based algorithm like genetic algorithms using similar operators; crossover, mutation and selection. In this thesis, a new way of selective image encryption scheme has been proposed which utilizes the main steps of Differential Evolution (DE) approach: crossover and mutation on the DCT coefficients. These two operations will be explored in the following subsections.

### 2.3.1 Differential Evolution Overview

In the optimization process of a difficult task [5], the method of first choice will usually be a problem specific heuristics. These techniques using expert knowledge achieve a superior performance. If a problem specific technique is not applicable due to unknown system parameters, the multiple local minima, or non-differentiability, Evolutionary Algorithms (EAs) have the potential to overcome these limitations [5].

EAs are a class of direct search algorithms. A conventional direct search method uses a strategy that generates variations of the design parameter vectors. Once a variation is generated, the new parameter vector is accepted or not. The new parameter vector is accepted in the case it reduces the objective function value. This method is usually named the greedy search. The greedy search converges fast but can be trapped by local minimum. This disadvantage can be eliminated by running several vectors simultaneously. This is the main idea of Differential Evolution (DE) algorithm [5]. Differential Evolution (DE) algorithm has been applied to several engineering problems in different areas.

The algorithm mainly has three advantages; finding the true global minimum regardless of the initial parameter values, fast convergence, and using a few control parameters. Being simple, fast, easy to use, very easily adaptable for integer and discrete optimization, quite effective in nonlinear constraint optimization including penalty functions and useful for optimizing multi-modal search spaces are the other important features of DE [6].

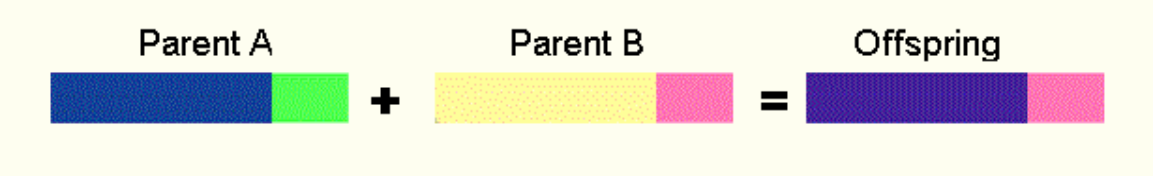
The DE algorithm is a population based algorithm like genetic algorithms using the similar operators; crossover, mutation and selection. The main difference in constructing better solutions is that the genetic algorithms rely on crossover while DE relies on mutation operation. This main operation is based on the differences of randomly sampled pairs of solutions in the population.

The algorithm uses mutation operation as a search mechanism and selection operation to direct the search towards the prospective regions in the search space. The DE algorithm also uses a non-uniform crossover that can take child vector parameters from one parent more often than it does from others. By using the components of the existing population members to construct trial vectors, the recombination (crossover) operator efficiently shuffles information about successful combinations, enabling the search for a better solution space.

### 2.3.2 Crossover Operation

Crossover selects genes from parent chromosomes and creates a new offspring. The simplest way how to do this is to choose randomly a crossover point and exchanges the subsequences before and after that point between two chromosomes to create two offsprings [5].

Crossover operation is denoted by the vertical bar symbol "|". For example in Figure 2.7, the strings 11011 | 00100110110 and 11011 | 11000011110 could be crossed over after the fifth locus in each to produce the two offsprings 11011 | 11000011110 and 11011 | 00100110110.



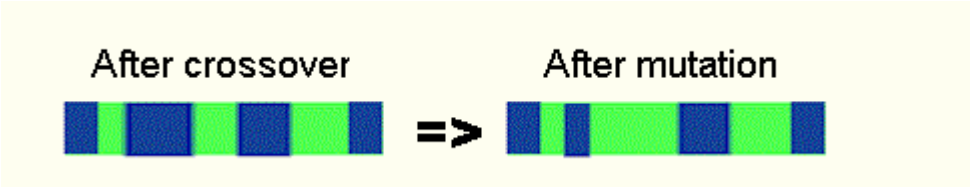

**11011 | 00100110110 +11011 | 11000011110=11011 | 11000011110**
**11011 | 00100110110 +11011 | 11000011110=11011 | 00100110110.**

Figure 2.7 Crossover example [5]

There are other ways how to make crossover, for example we can choose more crossover points. Crossover can be rather complicated and depends on the encoding of chromosome. Specific crossover made for a specific problem can improve performance of the DE algorithm. The crossover operator roughly mimics biological recombination between two single-chromosome (haploid) organisms.

### 2.3.3 Mutation Operation

After a crossover is performed, mutation takes place [5, 6]. This is to prevent falling all solutions in population into a local optimum of solved problem. Mutation changes randomly some of the bits in the new offsprings. For binary encoding we can switch a few randomly chosen bits from 1 to 0 or from 0 to 1. Mutation can then be as in Figure 2.8




**1101111000011110 => 1100111000011110**

Figure 2.8 Mutation example [5]

## 2.4 SELECTIVE ENCRYPTION OF IMAGES

It is argued that the traditional cryptosystems (such as RSA and DES), which have been originally developed for text data, are not suitable for securing many real-time multimedia applications because of large data sizes [17]. Software implementations of ciphers are usually too slow to process image and video data in commercial systems. Hardware implementations, on the other hand, add more cost to service providers and consumer electronics device manufacturers. So, the traditional cryptosystems need much time to directly encrypt the image data. The other problem is that the decrypted text must be equal to the original text. However, this requirement is not necessary for images; a decrypted image containing small distortion is acceptable due to human perception [18].

Because of the explosion of networks and the huge amount of content transmitted along, securing images content is becoming more and more important. A traditional approach for content access control is to first encode data with a standard compressor and then to perform full encryption of the compressed bitstream with a standard cipher (DES, AES, IDEA, etc.). In this scheme, called fully layered, compression and encryption are totally disjoint processes, as illustrated in Figure 2.9.

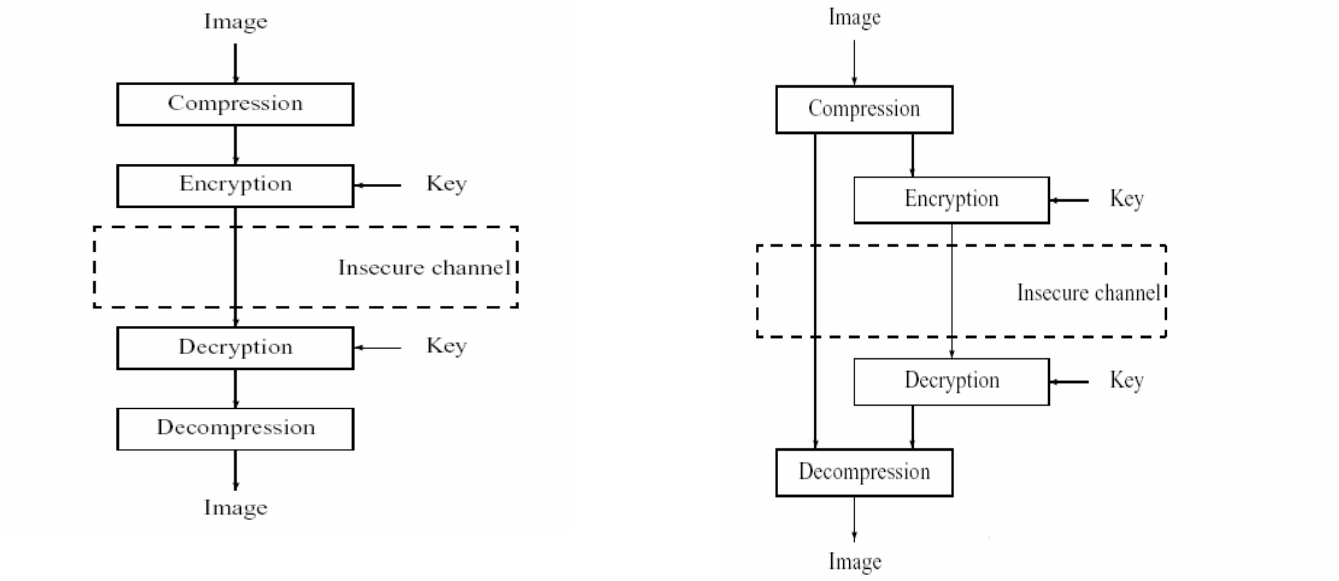



Figure 2.9 Compression-encryption systems a) Fully compression-encryption system: the whole compressed bitstreams of image is encrypted, and b) In perfect compression configuration, a subset of the bitstream of the image can be encrypted [18]

The media stream is processed as a classical text data with the assumption that all symbols or bits in the plain text are of equal importance. This scheme is relevant when the transmission of the content is unconstrained. In situations where only few resources are available (realtime networking, high-definition delivery, low memory, low power, or computation capabilities), this approach seems inadequate. Shannon [1] pointed out the specific characteristic of image and video content: high-transmission rate and limited allowed bandwidth, which justifies the inadequacy of standard cryptographic techniques for such content. Another limitation of the fully layered scheme consists of altering the original bitstream syntax. Therefore, many functionalities of the encoding scheme may be disabled (e.g., scalability). Some recent works explored a new way of securing the content, named, partial encryption or selective encryption, soft encryption, perceptual encryption, by applying encryption to a subset of a bitstream. The main goal of selective encryption is to reduce the amount of data to encrypt while achieving a required level of security. An additional feature of selective encryption is to preserve some functionalities of the original bitstream (e.g., scalability). The general approach is to separate the content into two parts [3].

The first part is the public part, it is left unencrypted and made accessible to all users. The second part is the protected part; it is encrypted. Only authorized users have access to protected part. One important feature in selective encryption is to make the protected part as small as possible. How to define public and protected parts depends on the target application. In some applications (video on demand, database search, etc.), it could be desirable to encourage customers to buy the content. For this purpose, only a soft visual degradation is achieved, so that an attacker would still understand the content but prefer to pay to access the full-quality unencrypted content. However, for sensitive data (e.g., military images/videos, etc.), hard visual degradation could be desirable to completely disguise the visual content. The peak signal-to-noise ratio (PSNR) [19] is the common criterion used to evaluate visual degradation [20].

In summary, selective encryption comes as an alternative that aims at providing sufficient security [21] with an important gain in computational complexity and delays. This allows a variety of possible applications for selective encryption. Below, we give a set of potential applications as follows [22].

(a) Mobile communication

PDAs, mobile phones, and other mobile terminals are more and more used for multimedia communication (voice, image, video, etc.) while still requiring copyright protection and access control. Their moderate resolution, computational power, and limited battery life impose to make an effort in reducing the encryption computational complexity to save battery life, silicon area, and cost. Image and video content have lower value than banking information, for example. Thus, it is not necessary to encrypt the whole data. It would be enough to degrade content quality so that people would prefer to buy a full-quality version.

(b) Monitoring encrypted content

One can imagine a situation where the encrypted content itself is usable for monitoring. For example, in many applications such as military images, video surveillance (where

some faces have to be scrambled), and media audience, identifying a partially encrypted content without decryption can be desirable.

(c) Multiple encryptions

Efficient overlay of more than one encryption system within a single bitstream can be very desirable. In a scheme where a TV broadcaster using an encryption system that is proprietary of one supplier wants to introduce new encryption systems of new independent suppliers, he would like to optimize bandwidth use by avoiding duplicating every channel on the network. Selective encryption could be very helpful; only a small fraction of the channel is duplicated (the part that will be encrypted). Each duplicated part will go through one supplier equipment and be encrypted by its encryption system. The remaining part (the shared one) will be sent once in the network and in the clear. Sony's Passage system proposed for the US cablemarket is a concrete example of this application. This solution is particularly desirable when the suppliers are not willing to agree on a shared scrambling solution as done in DVB Simulcrypt.

(d) Transcodability/scalability of encrypted content

These are very desirable properties in image and video communication. Some compression algorithms such as JPEG-2000 allow natural transcodability/scalability thanks to its embedded-code nature. For some other algorithms, it is necessary to decompress and recompress at lower bitrate at intermediate routers of the transmission channel. When the content is fully encrypted, decryption, decompression, and recompression at lower bitrate and reencryption are needed at intermediate routers. It may also cause important transmission delays and defeat the security of the system since access to the encryption key is needed at the network nodes. Selective encryption could be a good response to this problem. Encrypting a small fraction of the content while sending the remainder in the clear allows transcodability and scalability without accessing the encryption keys; the basic part (needed by all users) is sent in the clear (unencrypted) while the encrypted enhancement part is sent only to authorized users who paid to access the full-quality content.

(e) Database search

Selectively encrypted content can be used as low-quality previews that are made public. These previews will be used as catalogs to select content and pay to be able to decrypt and view.

(f) Renewable security systems

In their eternal battle against pirates, digital rights management systems have to periodically update their technologies and equipment all along the network. Changing the whole infrastructure would be very costly. Selective encryption can avoid the burden of having to change a whole system. Because of computational complexity saving due to selective encryption, it is possible to move to software solutions which are less expensive and can be easily and economically updated.

## 2.5 PREVIOUS SELECTIVE ENCRYPTION SCHEMES

Several selective encryption methods have been proposed for images. A classification of the proposed schemes from the open literature is given in Table 2.3 [22]. We will give a brief description of the proposed schemes.

Table 2.3 Classification of selective encryption schemes [22]

| Type of data | Domain | Proposal | Encryption Algorithm | What is encrypted? |
|---|---|---|---|---|
| Image | Frequency domain | Cheng & Li, 2000 | No algorithm is specified. | Pixel and set related significance information in the two highest pyramid levels of SPIHT |
| | | Droogenbroeck & Benedett, 2002 | DES, Triple DES and IDEA | Bits that indicate the sign and magnitude of the non-zero DCT coefficients |
| | | Pommer & Uhl, 2003 | AES | Subband decomposition structure |
| | Spatial Domain | Cheng & Li, 2000 | No algorithm is specified. | Quadtree structure |
| | | Droogenbroeck & Benedett, 2002 | Xor | Least significant bitplanes |
| | | Podesser, Schmidt & Uhl,2002 | AES | Most significant bitplanes |

### 2.5.1 Selective Encryption in Frequency Domain

**Cheng & Li, 2000 [23]:** In general, wavelet compression algorithms based on zerotrees transmit the structure of the zerotree with the significant coefficients. The SPIHT algorithm, for example, transmits the significance of the coefficient sets that correspond to trees of coefficients. Among the many different types of bits generated by the SPIHT algorithm, the proposed partial encryption scheme encrypts only the significance information related to pixels or sets in the two highest pyramid levels in addition to the parameter n that determines the initial threshold. The quadtree image compression is not part of any common image compression standard. Hence, the proposed partial encryption scheme may have limited use in commercial applications, since it has lack of bitstream compliance problem.

**Droogenbroeck & Benedett, 2002 [24]:** In JPEG compression, the Huffman coder aggregates zero coefficients into runs of zeros and uses symbols that combine the run of zeros with magnitude categories for the non-zero coefficients that terminate the runs. These symbols are assigned 8-bit code words by the Huffman coder. The code words precede the appended bits that specify the sign and magnitude of the non-zero coefficients. In the proposed scheme, the appended bits corresponding to a selected number of AC coefficients are encrypted using DES. The DC coefficients are left unencrypted because it is argued that they carry important visible information and are highly predictable. Encryption of at least 4 or 5 of the least significant bitplanes [6] means that the algorithm's computational cost is at least 50% to 60% with respect of the total encryption, so this scheme has insignificant computational reduction with respect to total encryption.

**Pommer & Uhl, 2003 [25]:** The encoder chooses different decomposition schemes with respect to the wavelet packet subband structure for each image that needs to be protected. Classical best basis selection algorithm is not appropriate to determine a useful wavelet packet basis as it results in trees that share common features for many images, leading to a potential security weakness. Instead, the generation of the decomposition tree is randomized using a pseudo random number generator (PRNG). The tree carrying the subband decomposition structure is then secured for transmission with AES encryption.

The amount of data to be encrypted for a given image is extremely small as no image data (e.g., transform coefficients) needs to be protected, so this scheme has insufficient security problem. Also, it decreases the compression performance of entropy coding, since the random generation of decomposition trees may not result in optimal compression efficiency when compared with wavelet packet coders targeted for image compression.

### 2.5.2 Selective Encryption in Spatial Domain

**Cheng& Li, 2000 [23]:** Quadtree image compression produces two logical parts: the quadtree and the parameters describing each block in the tree. The only parameter used by the authors to describe each block is the average intensity. As each intensity corresponds to a leaf node in the quadtree, the block intensities are called the leaf values. In the proposed partial encryption scheme, only the quadtree structure is encrypted. It can be used for both lossy compression (where each leaf is represented by the same number of bits) and lossless compression (where the number of bits to represent each leaf is different). For the transmission of the leaf values, two orderings are introduced: Leaf Ordering I (in order traversal of the quadtree) and Leaf Ordering II (the leaf values are encoded one level at a time from the highest level to the lowest level). For security reasons, Leaf Ordering I is not recommended for lossy or lossless compression while Leaf Ordering II is reportedly secure for both. The quadtree image compression is not part of any common image compression standards. Hence, the proposed partial encryption scheme may have limited use in commercial applications, since it has lack of bitstream compliance problem.

**Droogenbroeck & Benedett, 2002 [24]:** The decomposition of a gray scale image into its 8 bitplanes shows that the highest bitplanes exhibit some similarities with the original image while the least significant bitplanes look random. To exploit this property, some of the least significant bitplanes are encrypted. It is observed that at least 4 or 5 bitplanes need to be encrypted before the degradation becomes visible [29], so this scheme has insufficient security problem.

**Podesser, Schmidt & Uhl 2002 [26]:** The gray scale image is decomposed into its 8 bitplanes and the most significant bitplanes are encrypted. After a number of experiments, it is observed that (1) the encryption of the most significant bitplane is not secure enough, (2) selectively encrypting 2 bitplanes is sufficient if severe alienation of the image data is acceptable, and (3) encryption of 4 bitplanes provides high confidentiality. Two types of ciphertext only attacks on bitplane encryption (replacement attack and reconstruction attack) show that encryption of the most significant bitplane is not secure enough; at least 4 bitplanes need to be protected. Encryption of 4 most significant bitplanes [8] will result in a computational cost of 50% of the encryption ratio, so this scheme has insignificant computational reduction with respect to total encryption.

In this research, we propose a novel method of selective encryption for color and gray images that is based on the encryption of (Discrete Cosine Transform) DCT coefficients using Differential Evolution (DE) approach [5] [6]. The new selective encryption scheme should overcome most of the drawbacks which negatively impact on realtime applications, such as:

1. Insufficient security

2. Decrease in the compression performance

3. Insignificant computational reduction with respect to total encryption

4. Lack of bitstream compliance

5. Increase in key size

6. Static definition of encrypted part

# CHAPTER 3: MODEL METHODOLOGY AND ARCHITECTURE

In this research, a new selective encryption approach for JPEG compressed and uncompressed images is proposed, which is based on encryption of Discrete Cosine Transform (DCT) coefficients using Differential Evolution (DE) operations (crossover and mutation).

## 3.1 JPEG COMPRESSION PROCESS

The partial encryption scheme basically depends on a compression algorithm that decomposes the input image into a number of different logical parts. The output consists of some parts that provide significant amount of information about the original image, referred to as the important parts. The remaining parts have little meaning without the important parts, hence known as the unimportant parts. In this partial encryption approach, only the important parts need to be encrypted by a secure encryption algorithm. When the important part is considerably smaller than the total output of the compression, the encryption and decryption time can be reduced significantly [27].

There is a wide spectrum of JPEG applications that demand security on a modest level. Therefore, the search for fast encryption procedures appropriated for particular environments are required. Selective encryption is a recent approach to reduce the computational requirements for huge volumes of multimedia data in distribution networks with different client device capabilities [4]. It only protects the most important parts of an image to minimize computational efforts in real-time applications. Therefore, we concentrate on the JPEG standard which is more likely to be used in point to point transmission.

JPEG compression standard is discussed in details previously in Section 2.3 of this thesis. The next step of our model is block reorganization which is discussed in the following section.

## 3.2 BLOCKS REORGANIZATION FOR ENCRYPTION

After quantization step of JPEG compression process, recombine the resulted blocks in 2-D array where their positions are the same as their positions in the original image. Then, quantized blocks in 2-D array will be grouped to form segments.

In this research, two different sizes of the segment are used (The first size equals 3x3 blocks , while the second one equals 4x4 blocks) .This is done to study and analyze the effect of segment size on the results. Figure 3.1 depicts one segment consisting of 3x3 blocks.

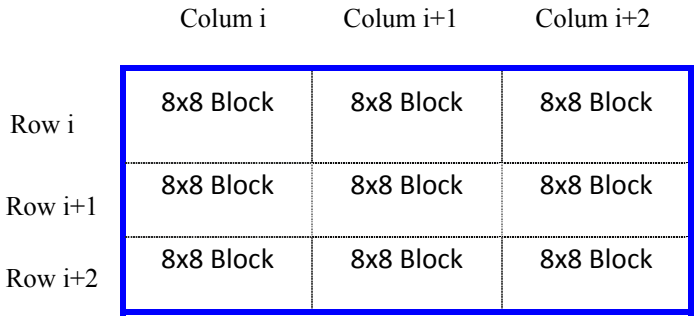



Figure 3.1 One segment consisting of 3x3 blocks

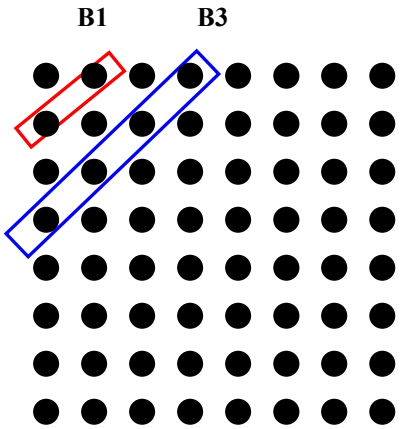
Due to the small block size in the transform DCT coding, the global characteristics of the image cannot be reflected in each block transformation, and grouping a set of blocks to a segment will be more effective. As a result of what has been mentioned earlier, segment correlation is utilized in the crossover step instead of block correlation, which reduces the time spent in determining the two blocks which should be crossed.

The final step in reorganization is division of each block in the segment into 14 bands (B1-B14) as shown in Figure 3.2(a). This division represents a 14 level frequency decomposition of quantized DCT coefficients and is done by separable filtering along the vertical and the horizontal directions.

Each band represents selected spatial frequency information of the transformed coefficients as shown in Figure 3.2(b). The statistics of the coefficient distribution generally differ from band to band. In addition, because the coefficients of the bands

are arranged in the spatial arrangement of the original image, neighboring coefficient correlation exists, which can be exploited by a bitstream coder.

The goal here is to provide a coefficient encryption method that does not significantly destroy these statistical properties. This is necessary to limit the crossover operation between the two decided cross segments within the DCT coefficients of the same bands, and thus reduce the time spent in crossover operation.




(a)

| DC | AC1 | AC5 | AC6 | AC14 | AC15 | AC27 | AC28 |
|---|---|---|---|---|---|---|---|
| AC2 | AC4 | AC7 | AC13 | AC16 | AC26 | AC29 | AC42 |
| AC3 | AC8 | AC12 | AC17 | AC25 | AC30 | AC41 | AC43 |
| AC9 | AC11 | AC18 | AC24 | AC31 | AC40 | AC44 | AC53 |
| AC10 | AC19 | AC23 | AC32 | AC39 | AC45 | AC52 | AC54 |
| AC20 | AC22 | AC33 | AC38 | AC46 | AC51 | AC55 | AC60 |
| AC21 | AC34 | AC37 | AC47 | AC50 | AC56 | AC59 | AC61 |
| AC35 | AC36 | AC48 | AC49 | AC57 | AC58 | AC62 | AC63 |

(b)

Figure3.2 The division of 8x8 block into bands: a) The bands of the 8x8 image block, and b) DCT coefficients (DC/AC).

Each band will include some AC coefficients as shown in Table 3.1.

## 3.3 SELECTIVE ENCRYPTION PROCESS

A digital image-scrambling scheme should have a relatively simple implementation, amenable to low-cost decoding equipment and low-delay operation for real-time interactive applications. It should have a minimum adverse impact on the compressibility of the image. It should preferably be independent of the bitstream compression selected for the image, and allow compression transcoding /scalability without having to decrypt. It should provide good overall security, although it may also be preferable in some systems to allow non-authorized users a level of transparency, both to entice them to pay for full transparency, and to discourage code-breaking.

Table 3.1 AC coefficients for each band

| Band | AC coefficients |
|---|---|
| 1 | 1,2 |
| 2 | 3,4,5 |
| 3 | 6,7,8,9 |
| 4 | 10,11,12,13,14 |
| 5 | 15,16,17,18,19,20 |
| 6 | 21,22,23,24,25,26,27 |
| 7 | 28,29,30,31,32,33,34,35 |
| 8 | 36,37,38,39,40,41,42 |
| 9 | 43,44,45,46,47,48 |
| 10 | 49,50,51,52,53 |
| 11 | 54,55,56,57 |
| 12 | 58,59,60 |
| 13 | 61,62 |
| 14 | 63 |

The proposed encryption approach aims to meet the objectives outlined above, and has three major steps: crossover, mutation, and scaling operations, to increase the level of security according to the application requirements. Details of each step are illustrated in the following subsections.

### 3.3.1 De-Crossover Operation

This operation is done on the DCT transformed image in the frequency domain, where it is easier to identify what parts of the data are critical for security purpose. This allows providing different levels of security and transparency, and also makes it easy to identify

what parts of the data are not compressible. Notice that the selected data can be easily located in the frequency domain without incurring any processing overhead.

In this research, crossover operation is proposed for the encryption process. But, the question arises now is, "what is the level of components used by this operation?"

For example, if the crossing is done between DCT coefficients of different frequencies within each block, this will lead to retain most of the local 2-D statistics of the block, but only coefficients around the block boundary may be slightly affected. Therefore, the negative impact on subsequent statistical coding is very small, while the visual effect of the crossing on a decompressed encrypted image is dramatic. The global block crossing changes the high-level spatial configuration of the image frequency content, which is much harder for an attacker to analyze than the local crossing of coefficients of different frequencies as proposed in [12] where statistics of different frequency components can be exploited for an efficient attack [11].

Exchange or crossing the arrangement of coefficients in a transform coefficient map can provide effective security without destroying compressibility, as long as the crossing is done between coefficients of the same frequency level which does not destroy the low-entropy aspects of the map relied upon by the bitstream coder.

The crossing is applied on the segment level. Within each segment, DCT coefficients of the same band (frequency location) are crossing within the corresponding DCT coefficients of the same band of the other segment. Since coefficients are crossed within a same band of the segments of the same frequency location, the crossover process should not significantly degrade the statistics relied upon by a run-length coder. We can also encrypt the sign of some coefficients (AC coefficients or DC coefficients) of non-intra coded blocks to increase the level of security as done in the coming encryption step (mutation).

The process of selecting the segments for crossing operation is determined by segments correlation function. The segments are sorted according to their correlation, and to make encrypted image unrecognizable, it is usually enough crossing merely some low

frequency bands such as band1 to band4. These bands are selected based on the trail experiments and taking into account the trade of between the usefulness of the increasing number of the bands on the security and the computational cost which will be produced. The coefficients of band2, band3, band4, and band5 of each block in the highest correlated segment will be crossed with the corresponding bands coefficients of each block in the lowest correlated segments. The selected crossover block of the first segment is determined using sub-key S11, while selected crossover block of the other segment is determined using sub-key S12.

Notice that the correlation function is the fitness function of the crossover operation that determines the segments that should be crossed. Finally, save the order of sorted segments in a list and send it to the decrypted side.

This step is reversible. Both encryption and decryption sides use the same shared secret keys (sub-keys K1, and K2) as initial seeds to pseudo-random number generators (PRNG). Therefore, the same encrypted blocks will be selected in the decrypted side. Moreover, the decryption side will select the same segments which are selected in the encryption side based on the list which is sent from the encryption side. Finally, the crossover effect will be removed and the original components will be retrieved again.

### 3.3.2 Calculation of Segment's Correlation

Due to the small block size in the transform DCT coding, the global characteristics of the image cannot be reflected in each block transformation. As an example, blocks in a smooth region of the image consist of the highly correlated pixels, and the energy of each block is concentrated on a few lower frequency DCT coefficients. So, the correlation of neighboring blocks (segment) is preferred. If the segment correlation is utilized to decide which segments are selected for crossing, the visual degradation of the reconstructed image can be improved, and the time spent in the crossing will be reduced.

In this research, the segment (superblock) is 3x3 or 4x4 DCT blocks and each block is 8x8 pixels .The DCT coefficients in the same locations are picked up from each block in a segment and rearranged into the secondary blocks as shown in Figure 3.3.

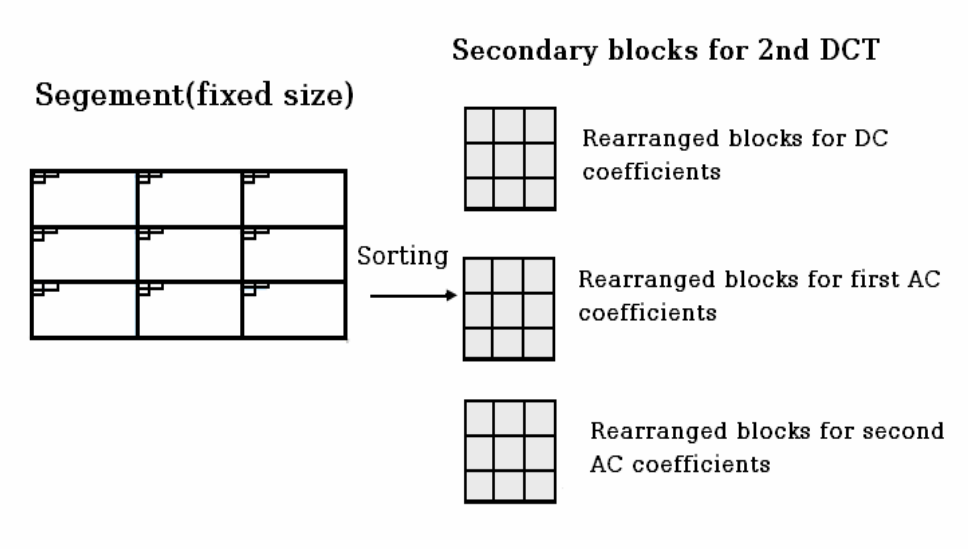



Figure 3.3 Division of the superblock (segment) to secondary blocks [28]

To calculate the segment correlation, the correlation value will be computed as [28]

$$\rho = \min\left[\frac{|r(0,1)|}{|r(0,0)|}, \frac{|r(1,0)|}{|r(0,0)|}\right] \tag{3.1}$$

where: $$r(k,l) = \frac{\sum_{m=0}^{N-1-k}\sum_{n=0}^{N-1-l} F'(m,n)F'(m+k,n+l)}{(N-k)(N-l)} \tag{3.2}$$

$$F'(m,n) = F(m,n) - \mu \tag{3.3}$$

$$\mu = \frac{1}{N^2}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} F(m,n) \tag{3.4}$$

Here $F(m,n)$, $\mu$ and *r (k, l)* are the element, mean and autocovariance of an NXN secondary block, respectively.

### 3.3.3 Watermarking (Hiding Information) Process

After crossover is done, we need to hide crossed segments' ranks in the encrypted image and send them to the decryption side. In this section, Watermarking by DC Coefficients Modification (WDCM) [29] algorithm is used to embed the secret information as a binary bit sequence in the quantized DC coefficients. The watermark embedding process can be applied in compression domain without re-encoding the data. Data mixing and extracting process of WDCM is explained in the following subsections.

#### 3.3.3.1 Data Mixing

Here is how we embed one bit of secret or watermark information into the quantized DC coefficient of an 8x8 block. Let $qDC_i$ be the quantized DC coefficient of block i, $b_i$ be the 1-bit information to be embedded in block i, and $n_i$ be the primary number which is uniformly distributed in [R1, R2]. The $qDC_i$ is modified according to [29]:

$$qDC_i' = \left\{ round\left[ n_i . round\left( \frac{qDC_i}{n_i} \right) \right], if \bmod \left[ round\left( \frac{qDC_i}{n_i} \right), 2 \right] \right\} = b_i \tag{3.5}$$

$$qDC_i' = \left\{ round\left\{ n_i . \left[ round\left( \frac{qDC_i}{n_i} \right) + \beta n_i \right] \right\}, if \bmod \left[ round\left( \frac{qDC_i}{n_i} \right), 2 \right] \neq b_i \right\} \tag{3.6}$$

Where:

$$\beta = \left\{ 1, \Rightarrow if \ldots qDC_i \geq n_i . round\left( \frac{qDC_i}{n_i} \right) or, round\left( \frac{qDC_i}{n_i} \right) = 0 \right\} \tag{3.7}$$

$$\beta = \left\{ -1 \Rightarrow if \ldots qDC_i < n_i . round\left( \frac{qDC_i}{n_i} \right) \right\} \tag{3.8}$$

The $\beta$ is chosen to be either –1 or 1 such that the modification process minimizes the absolute difference between the original $qDC_i$ coefficient and the $qDC_i$', as shown in

Equations (3.7) and (3.8). Since the quantized DC coefficients can be easily extracted in the JPEG bits stream, there is no need to decompress and re-compress the image for the proposed algorithm. Moreover, the AC coefficients are unaffected.

#### 3.3.3.2 Data Extraction

In order to decode the watermark, the PN sequence is needed. The watermark $b_i$ can be decoded according to:

$$b_i = \mathrm{mod}\left[ round\left( \frac{qDC_i}{n_i} \right), 2 \right] \tag{3.9}$$

### 3.3.4 De-Mutation Operation

Since it is easier in the frequency domain to identify what parts of the data are critical for security purpose, mutation operation is proposed to scramble the coefficients based on the recognition of a different characteristic of the transform coefficient data.

Although wholesale encryption of individual transform coefficients is generally undesirable (because coefficient encryption adds complexity and destroys the compressibility of the low-entropy coefficient data), some bits of individual transform coefficients have high entropy and can thus be encrypted without greatly affecting compressibility.

On the other hand, most transforms produce coefficients with sign bits that have an approximately equal probability of being a 1 or a 0, and that are highly uncorrelated with the sign bits of neighboring coefficients. The sign bits of the coefficients are usually difficult to compress; yet they are critical for security purpose. This uncompressible data segment can be selected for encryption without affecting the overall compression efficiency.

Because these bits have limited predictability to start with, scrambling them results in a negligible decrease in bitstream coding efficiency. So, we can mutate the sign of each AC coefficient in band1 to band6 in addition to DC coefficient. A Key-based cryptographically secure pseudo random process controls the sign change process. Note

that for many coders, the mean of the image is removed before the transform so that the coefficients in the lowpass band are also signed. Because the sign-inverted coefficients distribute their energy over the entire block of pixels they were derived from, sign bit mutation is quite effective at producing severe degradation in image quality, and the refinement bits of the coefficients can be scrambled too.

Mutation will be applied on n blocks of m segments, where m and n are inserted by the user to determine the degree of deterministic encryption he wants. The increasing in m, and n values, will lead to increase in the number of segments, and the number of blocks will be mutated at each segment and thus increase the level of security. S21 sequence generated from the sub-key K3 will be used to select the segment from the m ones, while S3 sequence generated from the sub-key K5 will be used to select the blocks of the selected segments from the n ones. Mutate the sign bits of the coefficients according to the mutate function as follows:

$$\text{Sign-bit Xor S22} \tag{3.10}$$

Where: S22 is generated from the sub-key K4.

This is justified by the fact that DCT sign bits are very random, thus neither predictable nor compressible, and for reducing computational complexity, we select the sign bits of the DCT coefficients to mutate.

Mutation operation is also reversible, because the same initial seed (sub-key K2) to the PRNG will be used in both encryption and decryption sides. The mutation effect will be removed by applying the mutation function again on the same components selected based on the generated indices.

### 3.3.5 Scaling DC Coefficients

The DC coefficients in the transformed blocks are easily identifiable though they are crossed and mutated to their huge magnitude compared to the AC coefficients. Due to this, an attacker can easily spot the DC coefficients and reconstruct an approximate image supplying constant values for the unknown AC coefficients. So, we propose to render the DC coefficients unidentifiable by scaling them using a constant sequence S3 derived from the secret key K.

Scaling DC coefficients is also reversible, because the same secret key K will be used in both encryption and decryption sides. The scaling effect will be removed by scaling the same components selected based on the generated indices with scaling factor 1/S3.

## 3.4 THE KEY GENERATION PROCEDURE

Since the security of the proposed encryption algorithm lies in the secret key, we propose a key generation technique in which an 80 bit user pass phrase K is split into five sub-keys K1, K2, K3, K4, and K5 of 16 bits each.

A cryptographically secure pseudo-random number generator (PRNG) is used to generate the sequences S11, S12 ,S21,S22,and S23 with seed values K1, K2 xor PRNG1 (shift register1 state), K3 xor PNRG2 state, K4 xor PNRG3 state, and K5 xor PNRG4, respectively. S11 and S12 sequences are used in crossover operation to choose which one of the blocks from the first segment will cross with which one from the other segment. S21, S22, and S23 are used in mutation operation. S22 is used for mutation function to xor the sign bit. S21 is used for selection of the segments to be mutated from the segments group, while S23 is used for selection of the blocks. S3 is derived from the secret key, and it is used to scale the DC coefficients. The proposed key generation process is depicted in Figure 3.4.

PRNG requires a naturally occurring source of randomness. Designing a hardware device or software program to exploit this randomness and produce a bit sequence that is free of biases and correlations is a difficult task. Additionally, for most cryptographic applications, the generator must not be subject to observation or manipulation by an

adversary [30, 31]. So, pseudorandom bit generator is used to create a sequence of bits that appears to be random. Linear feedback shift registers (LFSRs) are used in many of the keystream or bit sequence generators that have been proposed in the literature. There are several reasons for this [2]:

1. LFSRs are well-suited to hardware implementation,

2. They can produce sequences of large period,

3. They can produce sequences with good statistical properties, and

4. Because of their structure, they can be readily analyzed using algebraic techniques.

In this work, we used a LFSR that consists of 16-bits, with certain feedback function ($1 + x^{11} + x^{13} + x^{14} + x^{16}$), to give the pseudorandom bit sequence as shown in Figure 3.5.

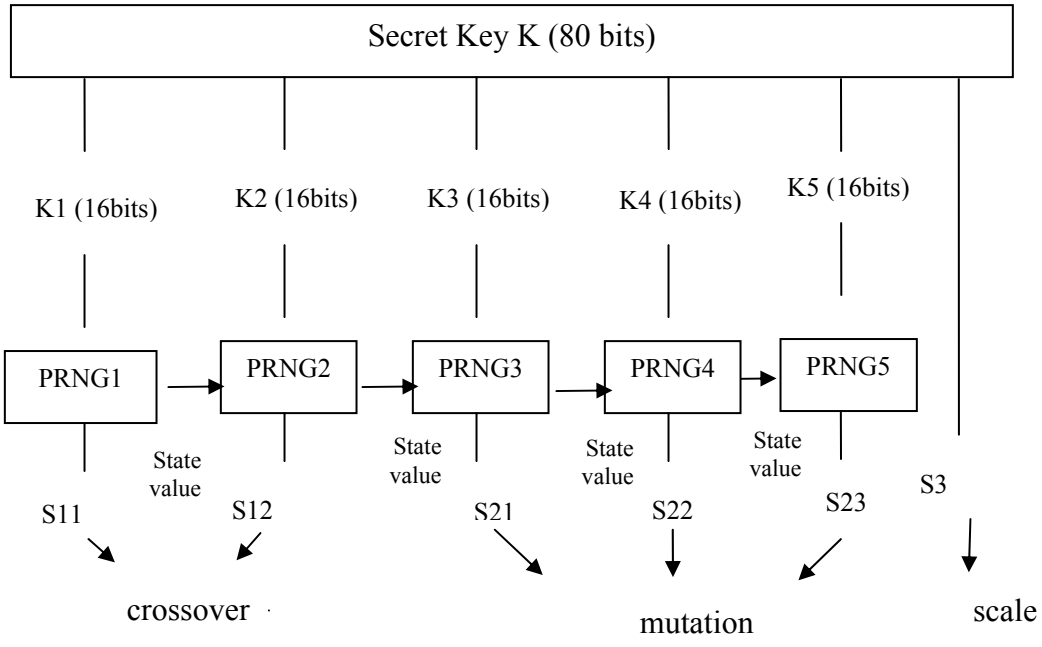



Figure 3.4 The key generation procedure

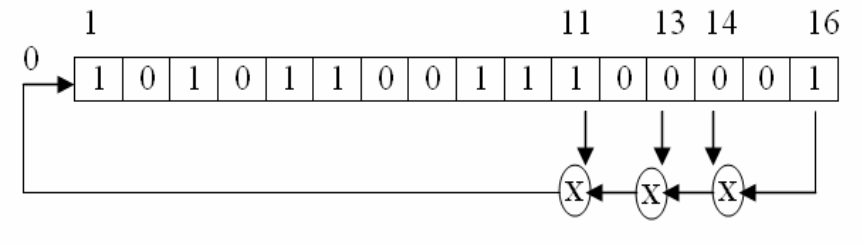



Figure 3.5 16-bit LFSR [30]

## 3.5 PROPOSED SELECTIVE ENCRYPTION/DECRYPTION WITH JPEG COMPRESSION

The steps involved in the proposed selective encryption process with JPEG compression are given below.

### 3.5.1 Selective Encryption Algorithm

**Input**: Original image of size HXW, secret key (80 bits)

**Output**: Encrypted - compressed image

**Step 1:** Read the image header, save the height of the image in variable **imageHeight** & the width in variable **imageWidth** and save the body of image in an array **imagBody**

**Step 2**: Obtain how many blocks exist in an image row and how many ones in the column, by dividing the width and height of the image by N, where N is equal to 8 (the required block size).

$$NoRowB = imageHeigh\ /N \quad (3.11)$$

$$NoColB = imageWidt\ /N \quad (3.12)$$

**Step 3:** For all blocks in the image perform the following:

- Get_block (row_no, col_no)

- Perform a DCT on the block by using Equation 2.3.
- Quantize the DCT coefficients in conjunction with a 64 element quantization table by using Equation 2.5.

**Step 4:**

- Reorganize the resulted blocks in an array **imagBlock**, as mention in Section 3.2.
- Divide **imagBlock** array to segments where each segment consists of 3x3 or 4x4 blocks.
- Divide each block in the segment into 14 bands (**B1-B14**).

**Step 5:** For all segments in the **imagblock** array perform the following:

***I. Crossover***

- Calculate the correlation for each segment in the **imgblock** using Equation 3.1, and then sort the segments according to their correlation. In this design, the correlation function is the fitness function that determines the bands that should be crossovered.
- Crossover coefficients of band2, band3, band4 and band5 of each selected block in the highest correlated segment with the corresponding bands coefficients of each selected block in the lowest correlated segments. The selected crossover block of first segment is determined using sub-key S11, while selected crossover block of the other segment is determined using sub-key S12.
- Repeat the previous step until crossing is done on all segments, and watermark the positions of the sorted segments in the resulted crossed coefficients using Equations 3.5 and 3.6 to send it to the decrypted side.

***II. Mutation***

- Enter the m, and n values to determine the number of the segments, and the number of the blocks in each segments, you need to mutate.

- Select m segments using S21 sequence generated using the sub-key **K3**, and selects the n blocks of the segments using S23 sequence generated using the sub-key **K5.** Then mutate the sign bits of the first 29 AC orders according to zigzag sequence, and the DC coefficients according to the mutate function, Equation 3.10 where S22 is generated from the sub-key **K3**.

***III. Scaling***

- Scale the magnitude of the DC coefficient using the scaling factor S3 sequence generated form the secret key K**.**

**Step 6:** Zigzag ordering and *Huffman coding*

- Order the AC coefficients in sequence using the zigzag ordering as mentioned in Subsection 2.2.5.

- Code the coefficients by using Huffman tables as the pair {HEAD, AMPLITUDE}.

The HEAD contains the controllers provided by the Huffman's tables. The AMPLITUDE is a signed-integer that is the amplitude of the nonzero AC, or in the case of DC is the difference between two neighbor DC coefficients. For the AC coefficients the HEAD is composed of (RUNLENGTH, SIZE), while for the DC coefficients it is made up only of SIZE.

**Step 7:** Reconstruct the image to get the encrypted one.

**Step 8:** End

### 3.5.2 Selective Decryption Algorithm

The decryption process is the reverse of the encryption process described above.

The steps involved in the proposed selective decryption process are given below.

**Input**: Encrypted-compressed image of size HXW, secret key, segPostion list

**Output**: Original image

**Step 1:** Read the image header, save the height of the image in variable **imageHeight** & the width in variable **imageWidth** and save the body image in an array **imagBody**

**Step2:**

- Decode the bitstream of the compressed data using the Huffman tables that were used during the compression process. The purpose of this step is to regenerate the zigzag ordered sequence of the quantized DCT coefficients.
- Reorder this zigzag sequence by the zigzag reordering step to create the 8x8 block of quantized DCT coefficients.

**Step 3:**

- Reorganize the resulted blocks in an array **imagBlock.**
- Divide **imagBlock** array to segments where each segment consists of 3x3 or 4x4 blocks.
- Divide each block in the segment into 14 bands (**B1-B14**).

**Step 4:**

For all segments in the **imagblock** array perform the following:

***III. AntiCrossover***

- Extract the watermarked positions of the sorted segments from the coefficients using Equation 3.9.

- Anticrossover coefficients of band2, band3, band4 and band5 of each selected block in the highest correlated segment with the corresponding bands coefficients of each selected block in the lowest correlated segments. The selected crossover block of first segment is determined using sub-key S11, while selected crossover block of the other segment is determined using sub-key S12.

- Repeat the previous step until anticrossing is done on all segments.

***IV. AntiMutation***

- Enter the m and n values to determine the number of the segments and the number of the blocks in each segments, you need to mutate.

- Select m segments using S21 sequence generated using the sub-key **K3**, and select the n blocks of the segments using S23 sequence generated using the sub-key **K5**. Then, mutate the sign bits of the first 29 AC orders according to zigzag sequence, and the DC coefficients according to the Mutate function Equation 3.10 where S22 is generated from the sub-key **K3**.

***III. AntiScaling***

- Scale the magnitude of the DC coefficient using the scaling factor 1/S3 sequence generated from the key K**.**

**Step5:**

- Dequantize the selected coefficients quantized in conjunction with a 64 element quantization table by using Equation 2.4.

- Perform an IDCT on the block using Equation 2.2.

**Step 6:** Reconstruct the image to get the plain one.

**Step 7:** End.

### 3.5.3 Model Flowchart

The flowchart of the proposed Selective Encryption/Decryption algorithm with JPEG compression is depicted in the Figure 3.6.

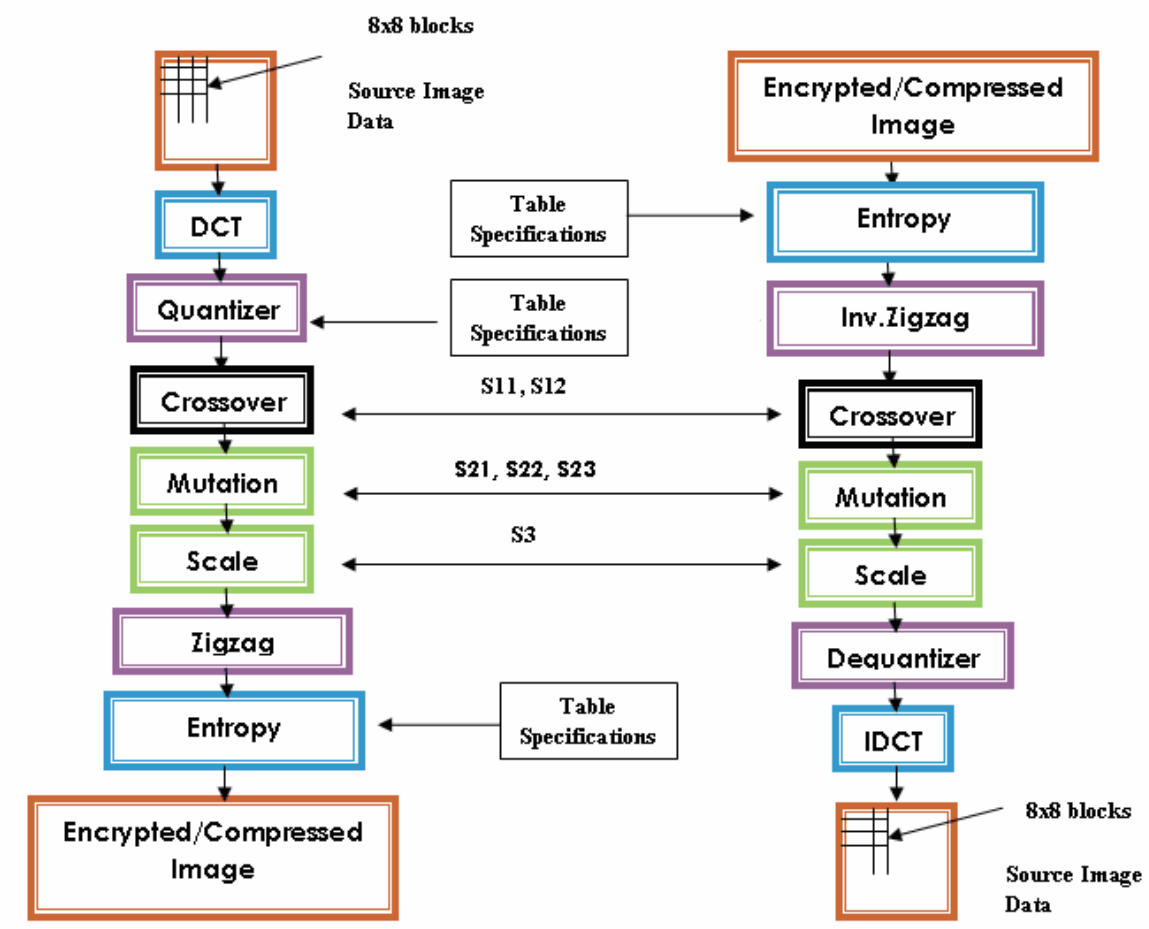



Figure 3.6 Encryption/decryption model with compression

## 3.6 PROPOSED SELECTIVE ENCRYPTION/DECRYPTION WITHOUT COMPRESSION

The previous schema of encryption can be used without doing encoding for images by replacing step 6 of the previous algorithm with:

- Dequantize the selected coefficients quantized in conjunction with a 64 element quantization table by using Equation 2.4.
- Perform an IDCT on the block using Equation 2.2.

Notice that the entropy coding process will not be needed in the uncompressed mode.

For the decryption side, change step 2 of the previous algorithm with:

- Perform a DCT on the block by using Equation 2.1.
- Quantize the DCT coefficients in conjunction with a 64-element quantization table by using Equation 2.3.

The flowchart of the proposed system is shown in Figure 3.7.

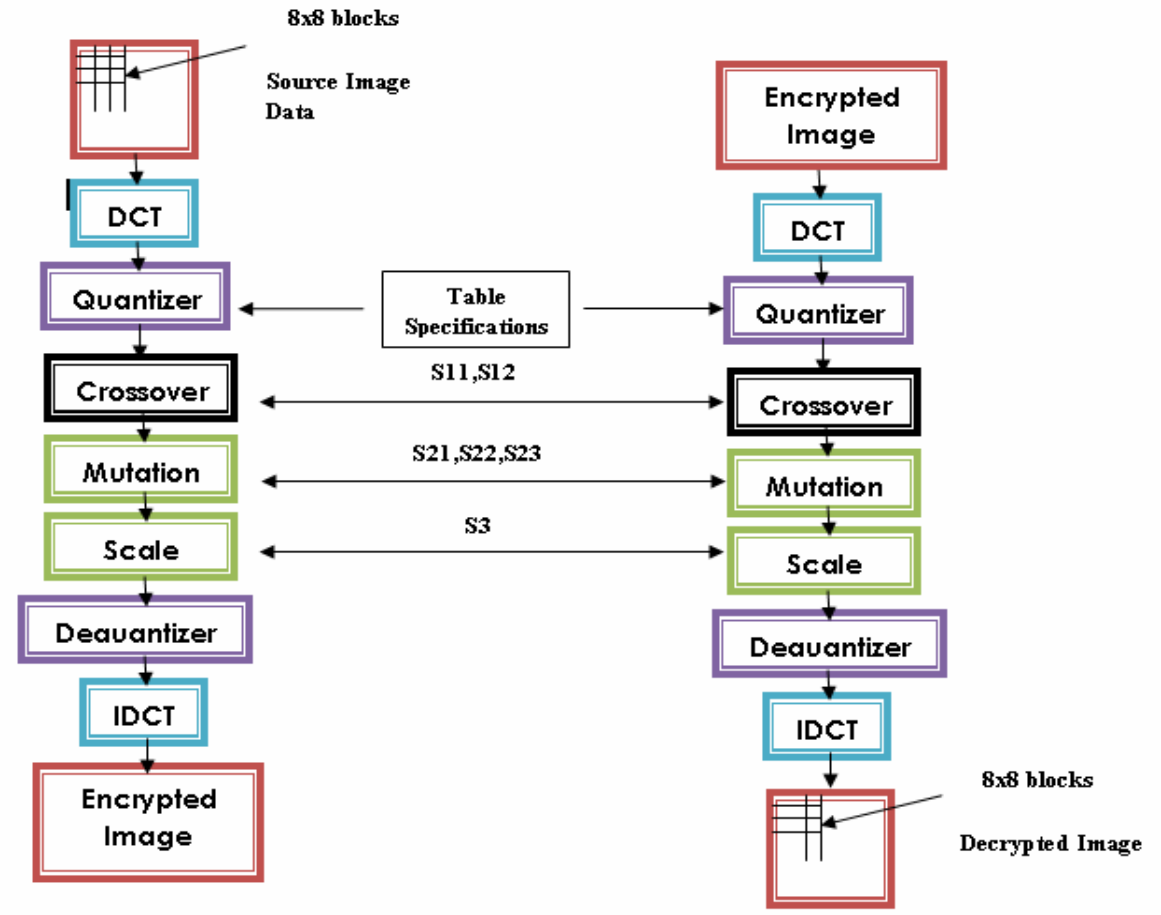



Figure 3.7 Encryption/decryption model without compression

# CHAPTER 4: SECURITY ANALYSIS AND TEST RESULTS

In this chapter, the performance of the proposed image encryption scheme is analyzed in detail. We discuss the security analysis of the proposed image encryption scheme including some important tests like visual testing, sensitivity analysis, and key space analysis, etc., to prove that the proposed cryptosystem has a suffienct level of security. For all of our experiments, we have used the JPEG algorithm for compression in baseline sequential mode with a quality factor (QF) of 50%. The algorithm is implemented using Matlab R2009b package.

## 4.1 VISUAL TESTING

A number of colored and gray scaled images are encrypted by the proposed method, and visual test is performed. Six examples of encryption are shown in Figure 4.1 (a) to Figure 4.6(a), with different images size (256x256, and 512x512 pixels), and with/without compression. By comparing the original and the encrypted images in Figure 4.1 to Figure 4.6, we notice that there is no visual information observed in the encrypted image, and the encrypted images are visual indistinguishable. Hence, this model succeeded in encrypting images of different sizes, and is suitable for small and large images as shown in the previous figures.

In additions, the visual indistinguishability of encrypted gray scaled images is more than that in colored images. This is due to the fact that, the gray scale images are composed of one plane (brightness /gray components), but the colored images are composed of the three planes one for the luminance (brightness/gray components) and two for the chrominance (color components), and the encryption process encrypts only the luminance (gray components) plane.

Besides that, encrypting the image with compression will give similar visual testing results comparing to encrypting without compression as shown in Figure 4.1(b) and 4.2(b). However the first case is more complex and takes longer time due to the compression process.

Let us change the m and n values used in the encryption (where m determines the number of the segments needed to mutate, while n equals the number of the mutated blocks in each segments), to study their effects on the visual test. The results of varying (m, n) to (4, 9), (4, 7), (4, 5), (2, 2),and (1, 1) are shown in the Figure 4.7 to Figure 4.11, where Sailboat image with compression is used and each segment includes 9 blocks. The m, and n values are selected based on the experimental trail tests. If m equals 4 then 50% of the segments will be used in the encryption process, if m equals 3 then 35% will be used, and if m equals 2 then only 20% of the segments will be used. The original image and its corresponding encrypted and decrypted images are shown in frames (a) to (c), and their histograms are shown in frames (d) to (f).

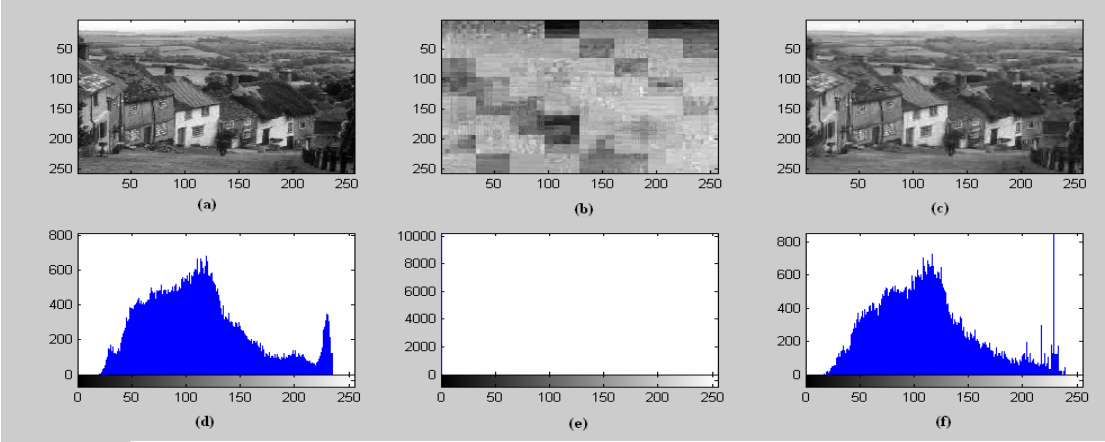



Figure 4.1 Encryption/compression 256x256 Hill standard images: (a) Original image, (b) Encrypted image, (c) Decrypted image, (d) Histogram of original image, (e) Histogram of encrypted image, and (f) Histogram of decrypted image.

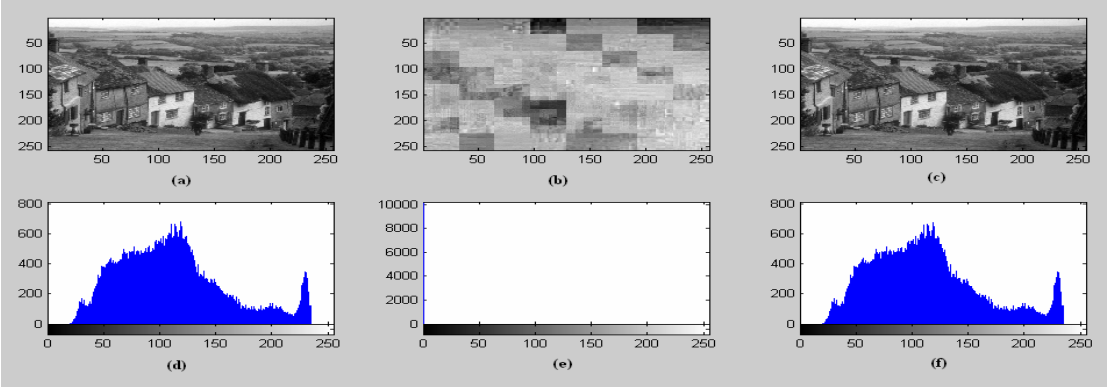



Figure 4.2 Encryption/without-compression 256x256 Hill standard images: (a) Original image, (b) Encrypted image, (c) Decrypted image, (d) Histogram of original image, (e) Histogram of encrypted image, and (f) Histogram of decrypted image.

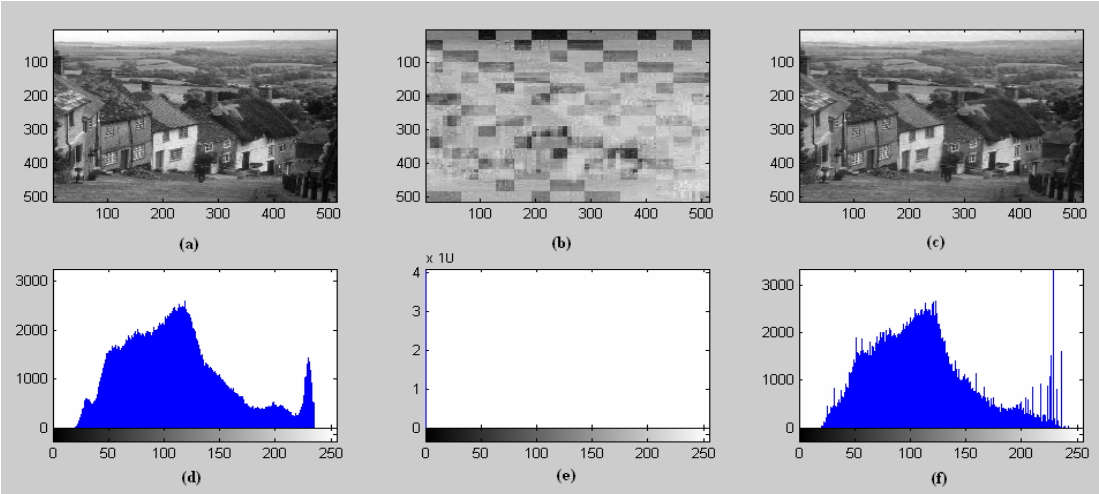



Figure 4.3 Encryption/compression 512x512 Hill standard images: (a) Original image, (b) Encrypted image, (c) Decrypted image, (d) Histogram of original image, (e) Histogram of encrypted image, and (f) Histogram of decrypted image.

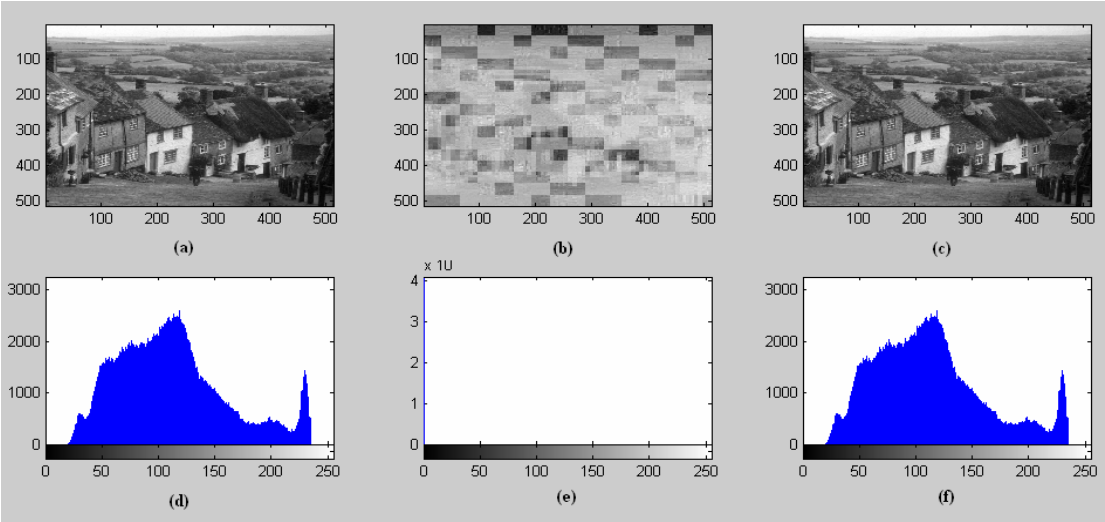



Figure 4.4 Encryption/without-compression 512x512 Hill standard images: (a) Original image, (b) Encrypted image, (c) Decrypted image, (d) Histogram of original image, (e) Histogram of encrypted image, and (f) Histogram of decrypted image.

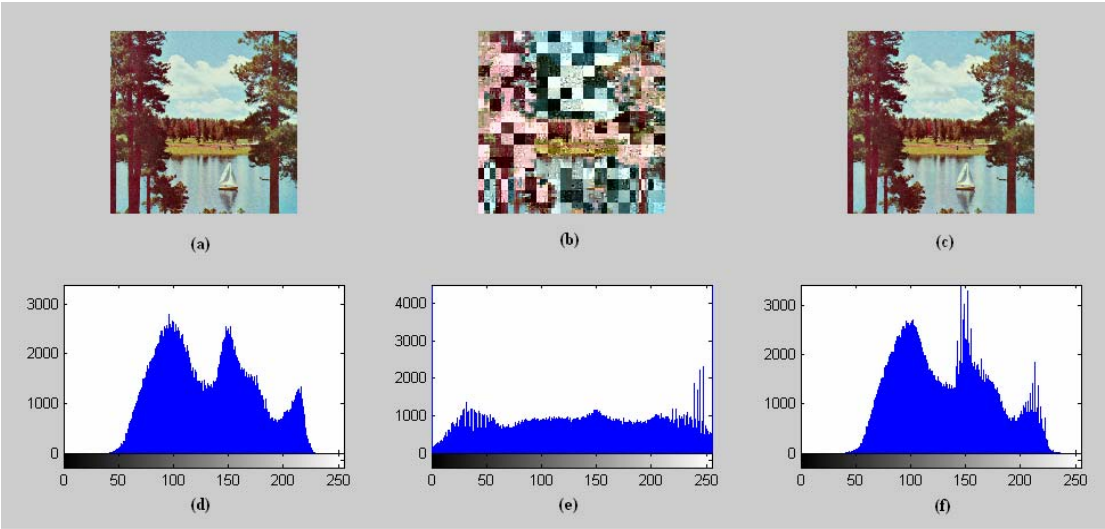



Figure 4.5 Encryption/compression 512x512 Sailboat standard images: (a) Original image, (b) Encrypted image, (c) Decrypted image, (d) Histogram of original image, (e) Histogram of encrypted image, and (f) Histogram of decrypted image.

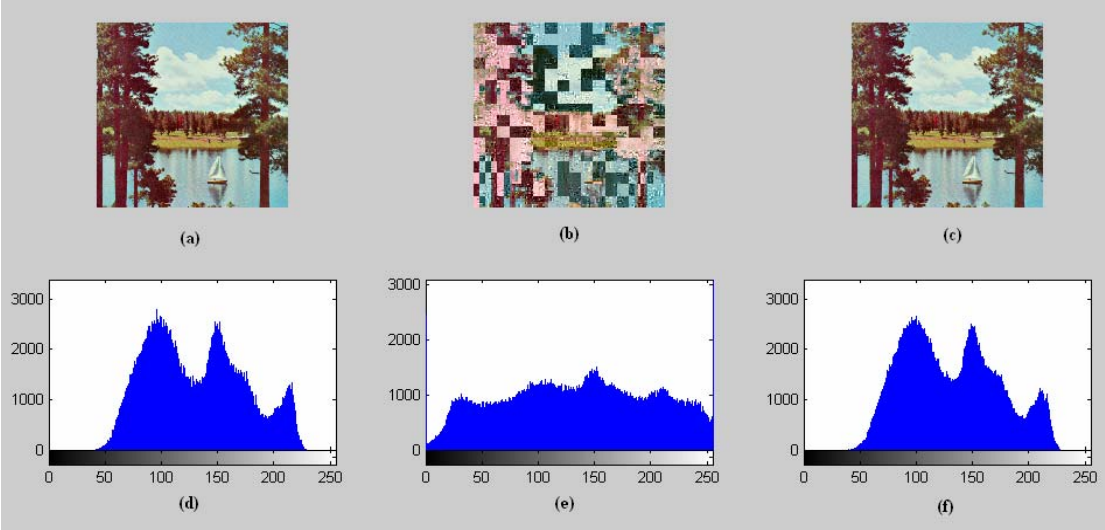



Figure 4.6 Encryption/without-compression 512x512 Hill standard images: (a) Original image, (b) Encrypted image, (c) Decrypted image, (d) Histogram of original image, (e) Histogram of encrypted image, and (f) Histogram of decrypted image.

Notice that, visual degradation is increased by increasing the values of m and n. The highest visual degradation is in Figures 4.7, where 50% of the segments in the image are considered in the encryption process since the security level m equals 4, and all of the 9 blocks in each segment is mutated, since the n equals 9. In contrast, more visual information is observed in Figures 4.11, where only one block of each segment is mutated

(n = 1) and 20% of the segments is considered in the encryption (m = 1). Thereby, this is successful method in offering different levels of security depending on the user application needs by controlling the security parameters (n and m values).

We increased the size of each segment to include 16 blocks and repeated the previous experiment for the Bridge image. The results are shown in the Figure 4.12 and Figure 4.16 with m and n values changed to (4, 16), (4, 10), (4, 14), (2, 6), (1, 1). Notice that the highest visual degradation is achieved in Figure. 4.12, and the lowest is achieved in Figure 4.16, this is due to the same reasons of the above experiment. In addition, if we increase the image size, similar results will be produced since the size does not have the significant effects on the results, and the proposed model is suitable for different sizes of the image.

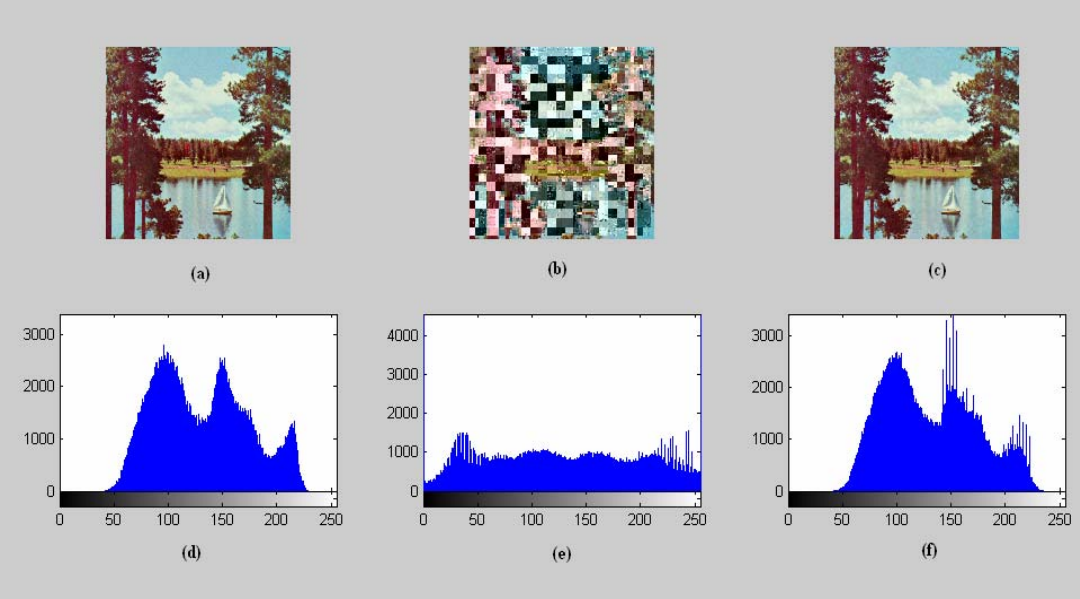



Figure 4.7 Encryption/compression 512x512 Sailboat standard images where m=4 and n=9: (a) Original image, (b) Encrypted image, (c) Decrypted image, (d) Histogram of original image, (e) Histogram of encrypted image, and (f) Histogram of decrypted image.

From the above experiments, the original images and their corresponding encrypted images are shown in the frames (a), (c), and their histograms are shown in frames (d), (e), It is clear that the histogram of the encrypted image is nearly uniformly distributed, and significantly different from the respective histograms of the original image. Hence, the

encrypted image does not provide any clue to employ any statistical attack on the proposed image encryption procedure, which makes statistical attacks difficult.

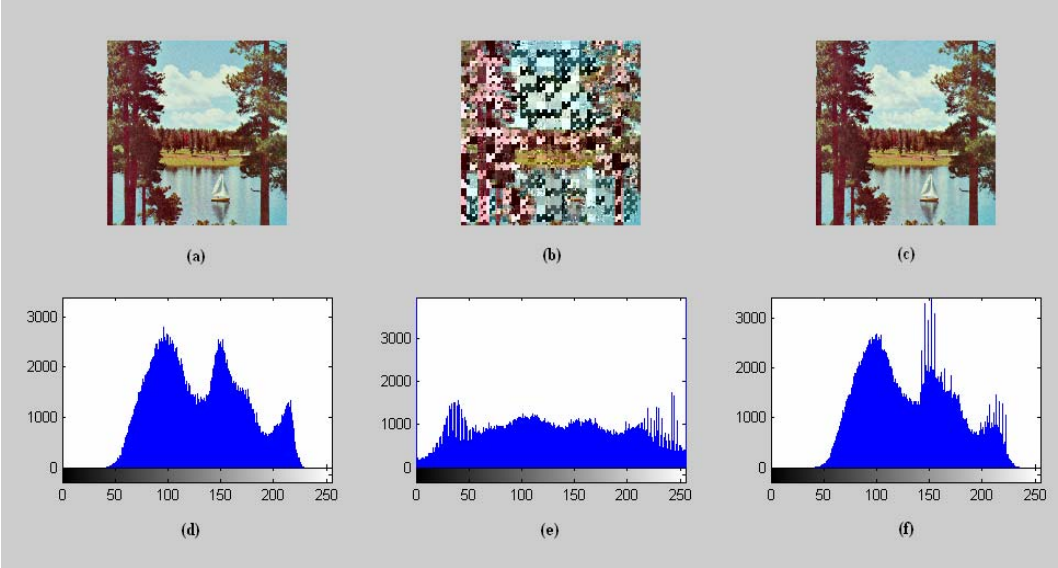



Figure 4.8 Encryption/compression 512x512 Sailboat standard images where m=4 and n=7: (a) Original image, (b) Encrypted image, (c) Decrypted image, (d) Histogram of original image, (e) Histogram of encrypted image, and (f) Histogram of decrypted image.

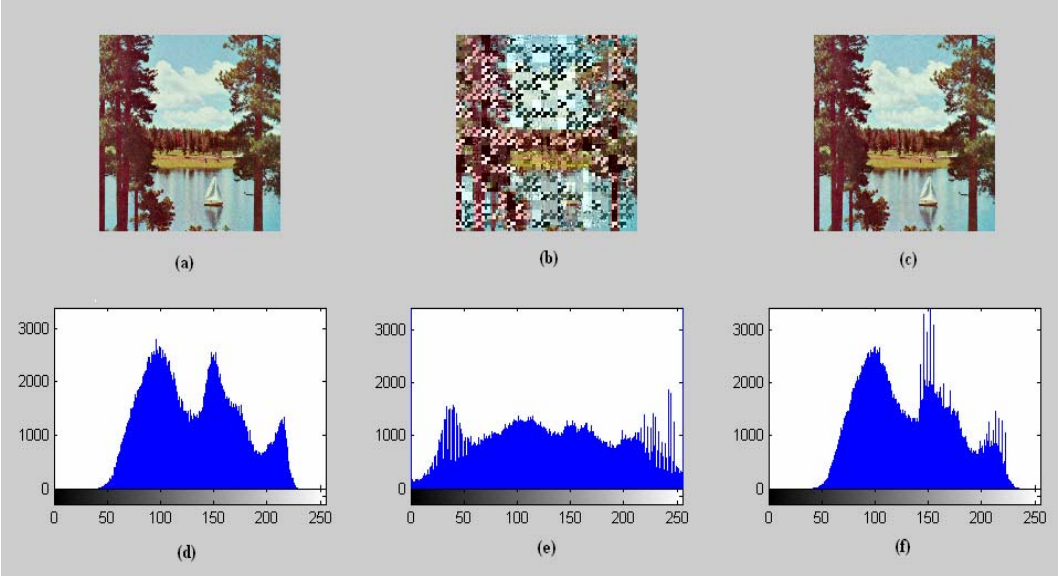



Figure 4.9 Encryption/compression 512x512 Sailboat standard images where m=4 and n=5: (a) Original image, (b) Encrypted image, (c) Decrypted image, (d) Histogram of original image, (e) Histogram of encrypted image, and (f) Histogram of decrypted image.

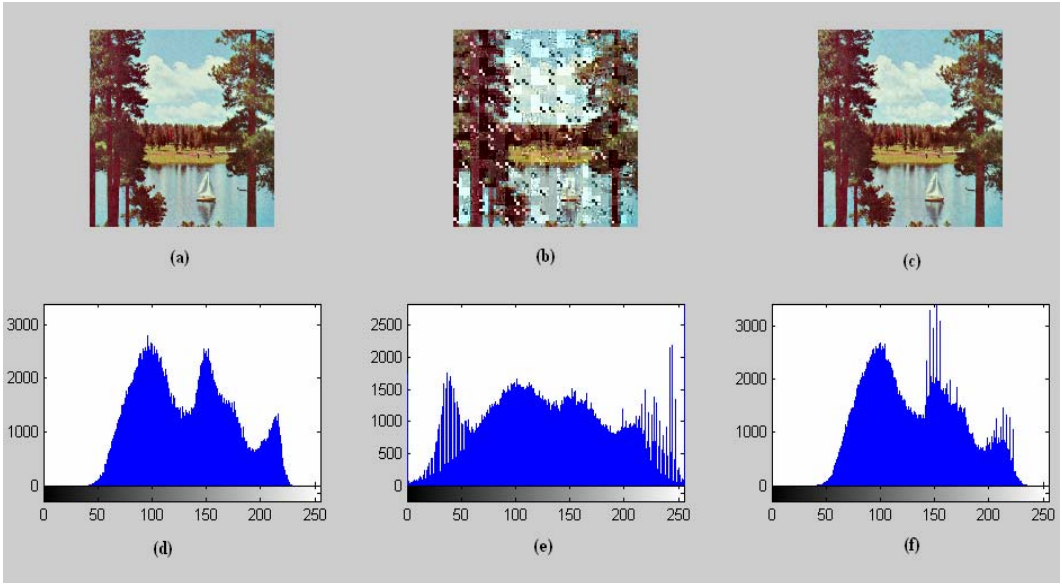



Figure 4.10 Encryption/compression 512x512 Sailboat standard images where m=2 and n=2: (a) Original image, (b) Encrypted image, (c) Decrypted image, (d) Histogram of original image, (e) Histogram of encrypted image, and (f) Histogram of decrypted image.

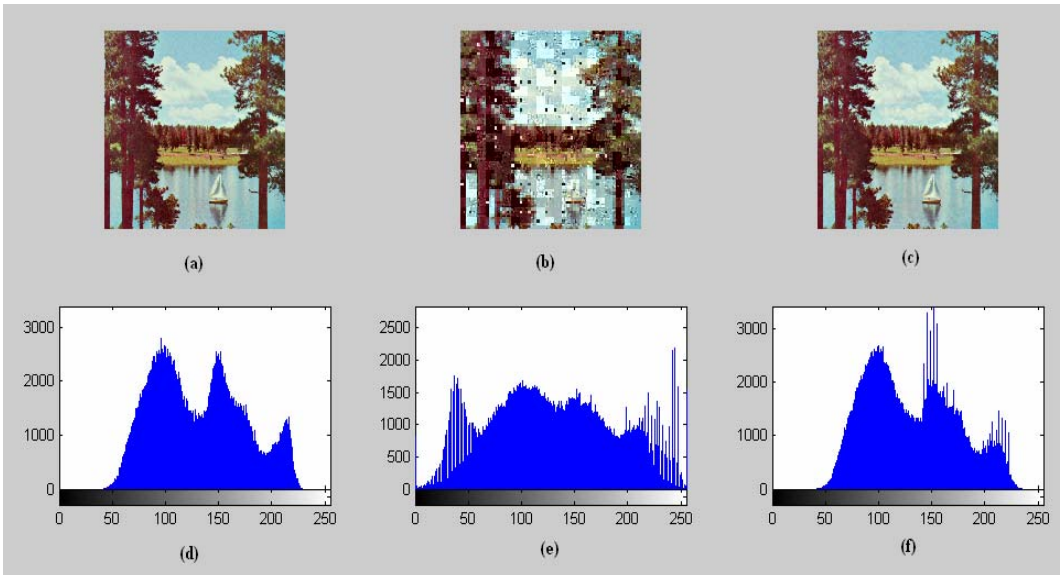



Figure 4.11 Encryption/compression 512x512 Sailboat standard images where m=1 and n=1: (a) Original image, (b) Encrypted image, (c) Decrypted image, (d) Histogram of original image, (e) Histogram of encrypted image, and (f) Histogram of decrypted image.

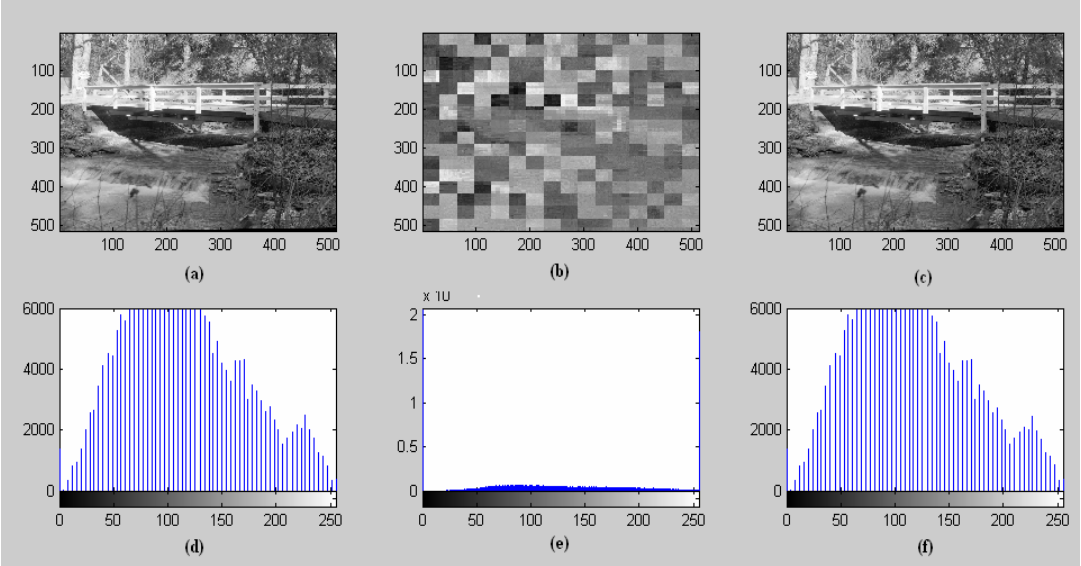



Figure 4.12 Encryption/without-compression 512x512 Bridge standard images where m=4 and n=16: (a) Original image, (b) Encrypted image, (c) Decrypted image, (d) Histogram of original image, (e) Histogram of encrypted image, and (f) Histogram of decrypted image.

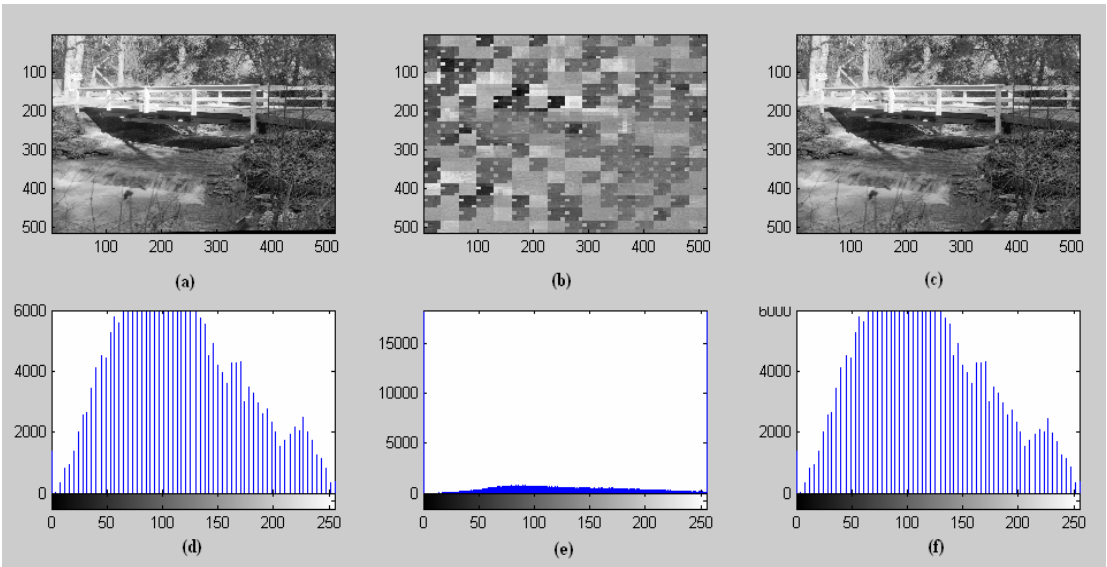



Figure 4.13 Encryption/without-compression 512x512 Bridge standard images where m=4 and n=14: (a) Original image, (b) Encrypted image, (c) Decrypted image, (d) Histogram of original image, (e) Histogram of encrypted image, and (f) Histogram of decrypted image.

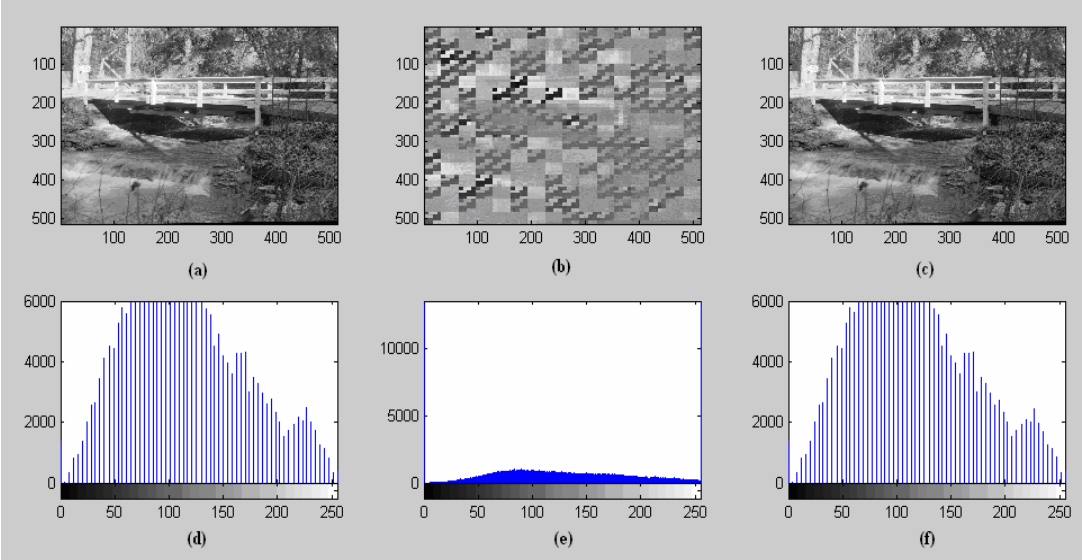



Figure 4.14 Encryption/without-compression 512x512 Bridge standard images where m=4 and n=10: (a) Original image, (b) Encrypted image, (c) Decrypted image, (d) Histogram of original image, (e) Histogram of encrypted image, and (f) Histogram of decrypted image.

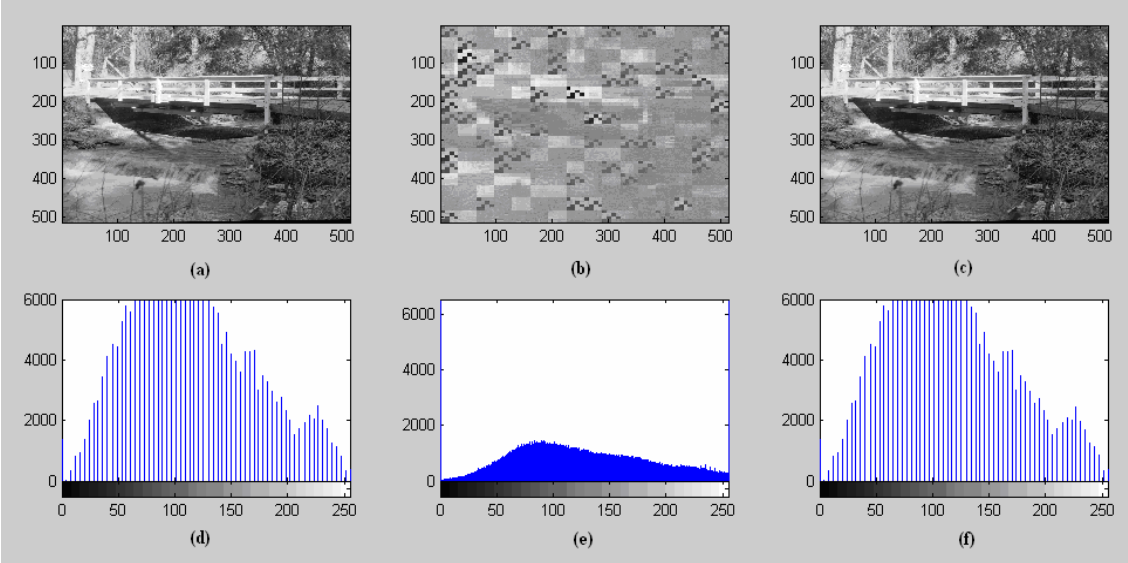



Figure 4.15 Encryption/without-compression 512x512 Bridge standard images where m=2 and n=6: (a) Original image, (b) Encrypted image, (c) Decrypted image, (d) Histogram of original image, (e) Histogram of encrypted image, and (f) Histogram of decrypted image.

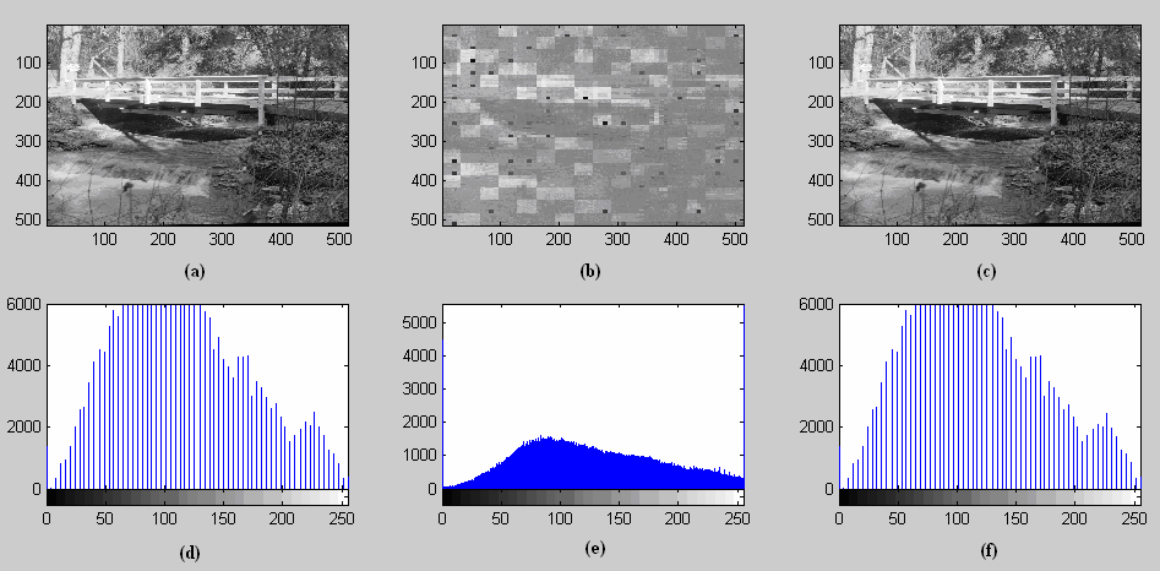



Figure 4.16 Encryption/without-compression 512x512 Bridge standard images where m=1 and n=1: (a) Original image, (b) Encrypted image, (c) Decrypted image, (d) Histogram of original image, (e) Histogram of encrypted image, and (f) Histogram of decrypted image.

Experiments on various images have shown similar results. These properties tell that the proposed cryptosystem has high security against statistical attacks. For example, in Sailboat plain image histogram shown in Figure 4.5(d), some values in the ranges [0, 50], and [230, 250] are missing, but they exist and are uniformly distributed in the encrypted image histogram Figure 4.5(e). Different images have been tested by the proposed image encryption procedure, and missing some ranges in the encrypted histogram are obtained, which also shows that the proposed key-generation procedure can generate sequences with good pseudo-random properties

The peak signal-to noise ratio (PSNR) [32] is the main metric used in the literature to measure visual degradation. Visual degradation is a subjective criterion that is why it is difficult to define a threshold for acceptable visual distortion regarding a given application.

It is most easily defined via the mean squared error (MSE) which for two $m \times n$ images $I$ and $K$, where one of the images is considered a noisy approximation of the other, is defined as:

$$MSE = \frac{1}{mxn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \left[ I(i,j) - K(i,j) \right]^2 \tag{4.1}$$

The PSNR is defined as:

$$PSNR = 10x \log_{10} \left( \frac{MAXI^2}{MSE} \right) \tag{4.2}$$

Here, MAXI is the maximum possible pixel value of the image.

The test results for compression images are shown in Figures 4.17 and 4.18. Notice that, with the increase of security level parameters (m, n), the PSNR between the original image and the encrypted one is decreased, since the visual degradation is increased.

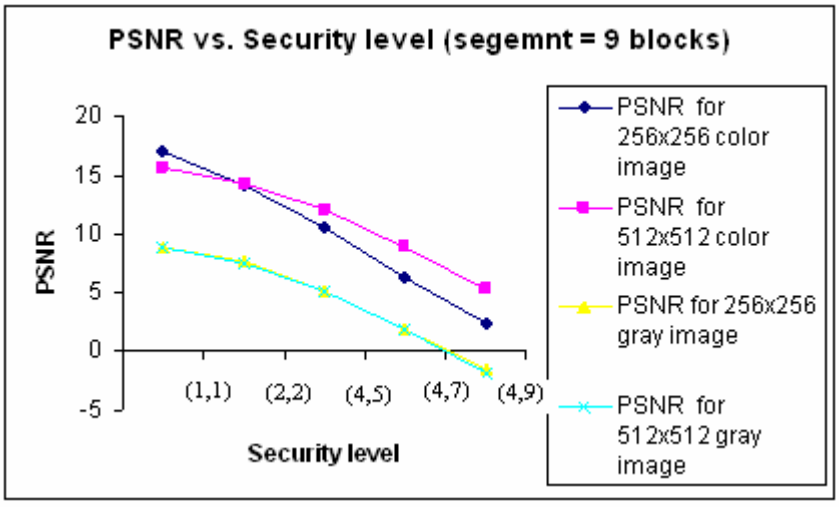



Figure 4.17 PSNR vs. security level when the segment includes 9 blocks

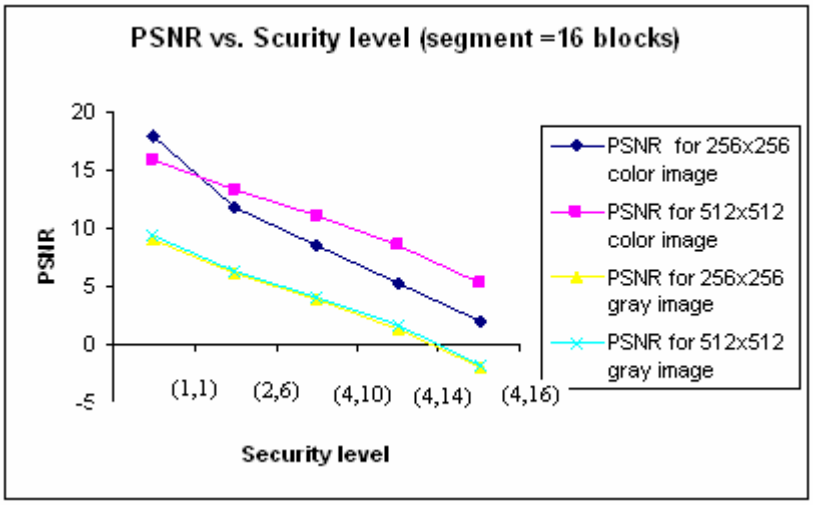



Figure 4.18 PSNR vs. security level when the segment includes 16 blocks

Besides that, PSNR for the gray scaled image is less than the colored image, since more visual information can be observed in the color one for the same reason illustrated above in visual testing analysis.

In order to further demonstrate the effectiveness, some more sophisticated tests have been carried out and the results are to be explained in the following.

### 4.2 Statistical Analysis

In order to resist the statistical attacks, the encrypted images should possess certain random properties. We have performed statistical analyzed by calculating the histograms, and the correlation coefficient for several images and its corresponding encrypted images. Since the histogram is previously analyzed in the above section, a detailed study of correlation coefficient analysis has been performed and the results are summarized as following. Different images have been tested, and similar results are obtained. However, due to the page limit, only the results for the Hill, (Figure 4.1(a)), the Sailboat (Figure 4.5(a)), and the Bridge (Figure 4.12(a)), are used for illustration.

### 4.2.1 Correlation Coefficient Analysis

In order to test the correlation between the image and its corresponding encrypted image by using the proposed encryption algorithm, the following two formulas are used to calculate the correlation [34]:

$$\mathrm{cov}(x, y) = E(x - E(x))(y - E(y)) \tag{4.3}$$

$$r_{xy} = \frac{\mathrm{cov}(x, y)}{\sqrt{D(x)}\sqrt{D(y)}} \tag{4.4}$$

Where x and y are values of two correlating pixels in the original image and the encrypted image. In numerical computation, the following discrete formulas are used:

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{4.5}$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N} (x_i - E(x))^2 \tag{4.6}$$

$$\mathrm{cov}(x, y) = \frac{1}{N}\sum_{i=1}^{N} (x_i - E(x))(y_i - E(y)) \tag{4.7}$$

The results in Table 4.1 illustrate that the average correlation coefficients of sets of colored images are very small which implies that no correlation exists between the original and its corresponding encrypted image, and its value is decreased by the increase of the security level's parameters (n, m). As shown in the table, smallest value of the correlation is achieved at n = 16 m = 5, when the segment consists of 16 blocks, and n = 9 m = 5 when the segment consists of 9 blocks.

Table 4.1 Correlation coefficient between plain and encrypted images.

| Num. of blocs. in seg. | Image Size | Security level | | | | |
|---|---|---|---|---|---|---|
| | | Lowest | low | medium | high | highest |
| 9 | 128x128 | 0.633 | 0.5869 | 0.2472 | 0.1291 | -0.6535 |
| | 256 x256 | 0.5827 | 0.4447 | 0.1078 | 0.0368 | -0.2268 |
| | 512x512 | 0.7078 | 0.6231 | 0.1712 | -0.0086 | -0.7759 |
| | 1024x1024 | 0.8634 | 0.8113 | 0.4683 | 0.2894 | -0.7232 |
| 16 | 128x128 | 0.5188 | 0.3714 | -0.0292 | -0.195 | -0.6852 |
| | 256 x256 | 0.6402 | 0.3367 | 0.0954 | 0.0265 | -0.5246 |
| | 512x512 | 0.7323 | 0.5437 | 0.1131 | -0.0875 | -0.7922 |
| | 1024x1024 | 0.881 | 0.7663 | 0.4477 | 0.2662 | -0.7673 |

### 4.3 Sensitivity Analysis

An encryption scheme has also to be key-sensitive, meaning that a tiny change in the key will cause a significant change in the output. In our tests, the key sensitivity test is performed in detail according to the following steps [33]:

1. First, the images in the figures are encrypted by using the test key K1 = "125462541222565441236 6" (in decimal), and their corresponding encrypted images are referred as encrypted images A (frames (b) in the figures), where and n=4 and m=10 for 16 blocks segments.

2. Then, the least significant bit of the key is changed, so that the new key becomes, K2 = ''125462541222565441236 7''. This key is used to encrypt the same images, and their

corresponding encrypted images are referred as encrypted images B (frames (c) in the figures).

3. Finally, the correlation among the original images, the encrypted images A, and the encrypted images B are computed, and the results are shown in Table 4.2 and Figures 4.21 to 4.26.

Table 4.2 Correlation coefficient among plain images, encrypted A images, and encrypted B images

| Image Type | Segment size | Image Size | Correlation original/A | Correlation original/B | Correlation A/B |
|---|---|---|---|---|---|
| Gray scale | **9** | 256 x256 | -0.0342 | -0.0533 | 0.1281 |
| | **16** | 256x256 | 0.0014 | -0.1785 | 0.2112 |
| Colored | **9** | 256 x256 | -0.0386 | -0.1167 | 0.0259 |
| | **16** | 256x256 | 0.0031 | 0.147 | 0.0848 |

In Table 4.2, the correlation between plain images and the corresponding encrypted images is negligible around zero, which shows that the plain image is nearly independent from the encrypted images. This is consistent with the perfect security defined by Shannon [5].

Similarly, the correlation between different encrypted images A and B is also negligible around zero when the security level is high, which shows that the encrypted images are independent from each other.

Moreover, if we use a trivially modified key K2 to decrypt the ciphered images A, then the decryption should not succeed. In Figures 4.19, 4.20, the compressed images are encrypted, but in Figures 4.21, and 4.22, the compression is not used. Frames (d) in these figures have verified that the images encrypted by the K1 were not be correctly decrypted

by using the K2. Here, there is only one bit difference between the keys explained before. Those results clearly show high key-sensitivity of the proposed encryption algorithm.

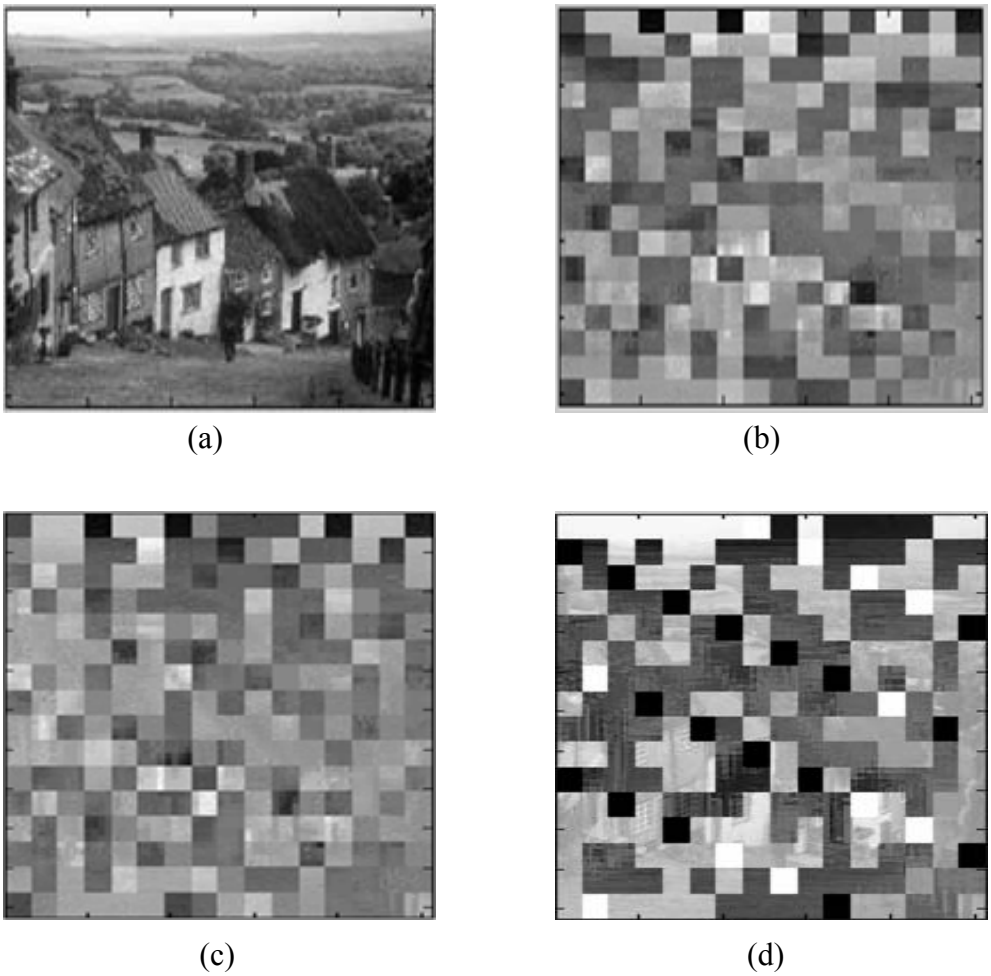



Figure 4.19 Key sensitivity test: (a) Original Gold Hill 512x512 image, (b) encrypted compressed image using K1, (c) encrypted compressed image using K2, and (d) decrypted image in (b) using wrong key K2.

### 4.4 Key Space Analysis

Key space size is the total number of different keys that can be used in the encryption. Cryptosystem is completely sensitive to all secret keys. A good encryption algorithm should not only be sensitive to the cipher key, but also the key space should be large enough to make brute-force attack infeasible. In this algorithm, the key consists of 80 bits; the key space size is over than $12\times10^{23}$. Apparently, the key space is sufficient for reliable practical use.

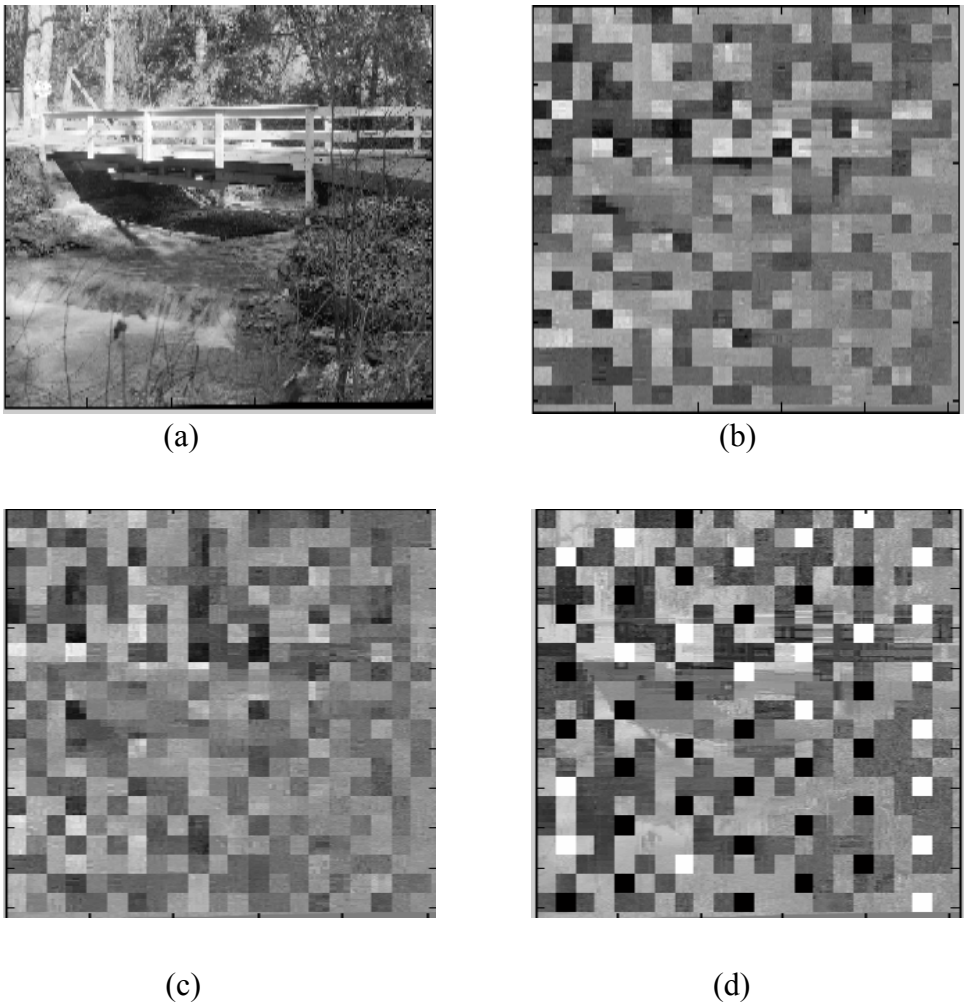
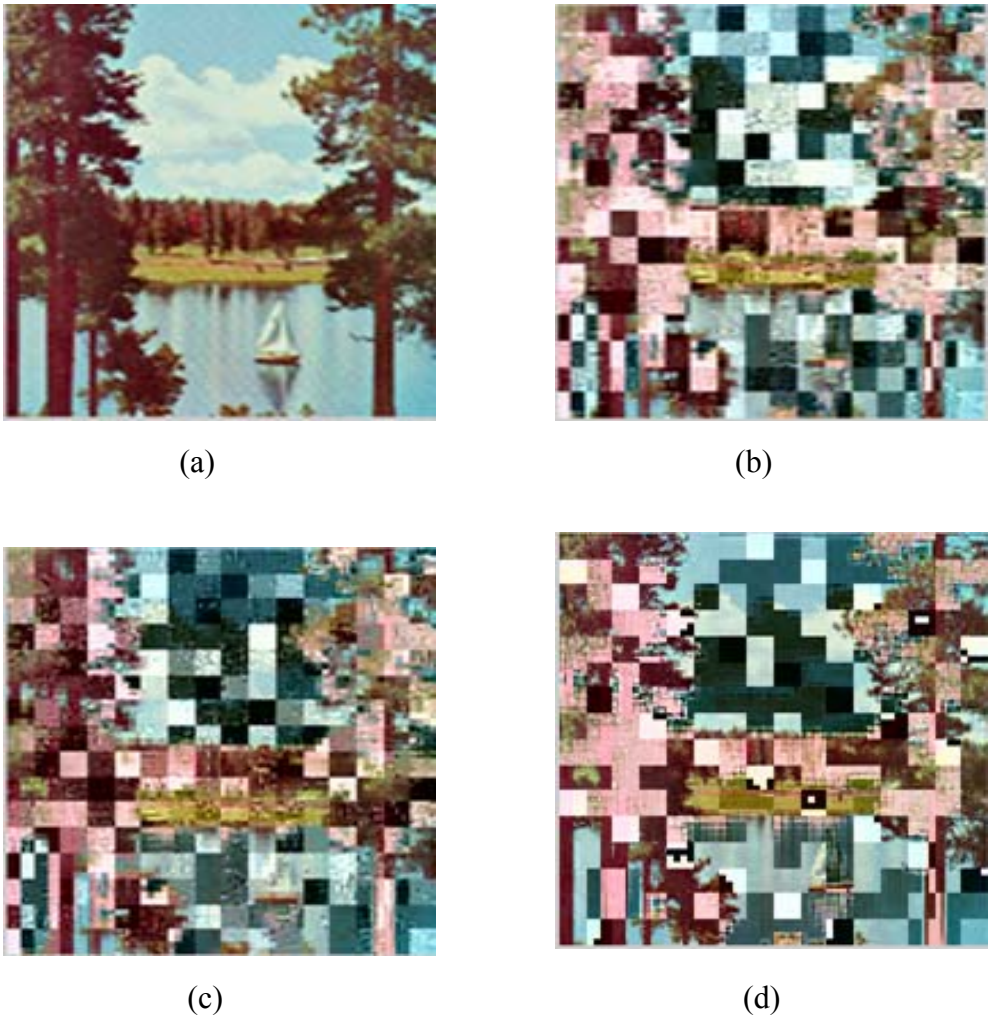


Figure 4.20 Key sensitivity test: (a) Original Bridge 512x512 image, (b) encrypted image using K1, (c) encrypted image using K2, and (d) decrypted image in (b) using wrong key K2.

### 4.5 Replacement Attack

It should be noticed that security is linked to the ability to guess the values of the encrypted data. For example, from a security point of view, it is preferable to encrypt the bits that look the most random. However, in practice this trade-off is challenging because the most relevant information, such as the DC coefficients in a JPEG encoded image are usually highly predictable [10]. In the experiment, we have replaced the encrypted DCT

coefficients with constant values. For example, we get the image illustrated in Figure 4.23(b), and 4.23(d) as the results of replacement attack of Figure 4.23(a), and 4.23(c).

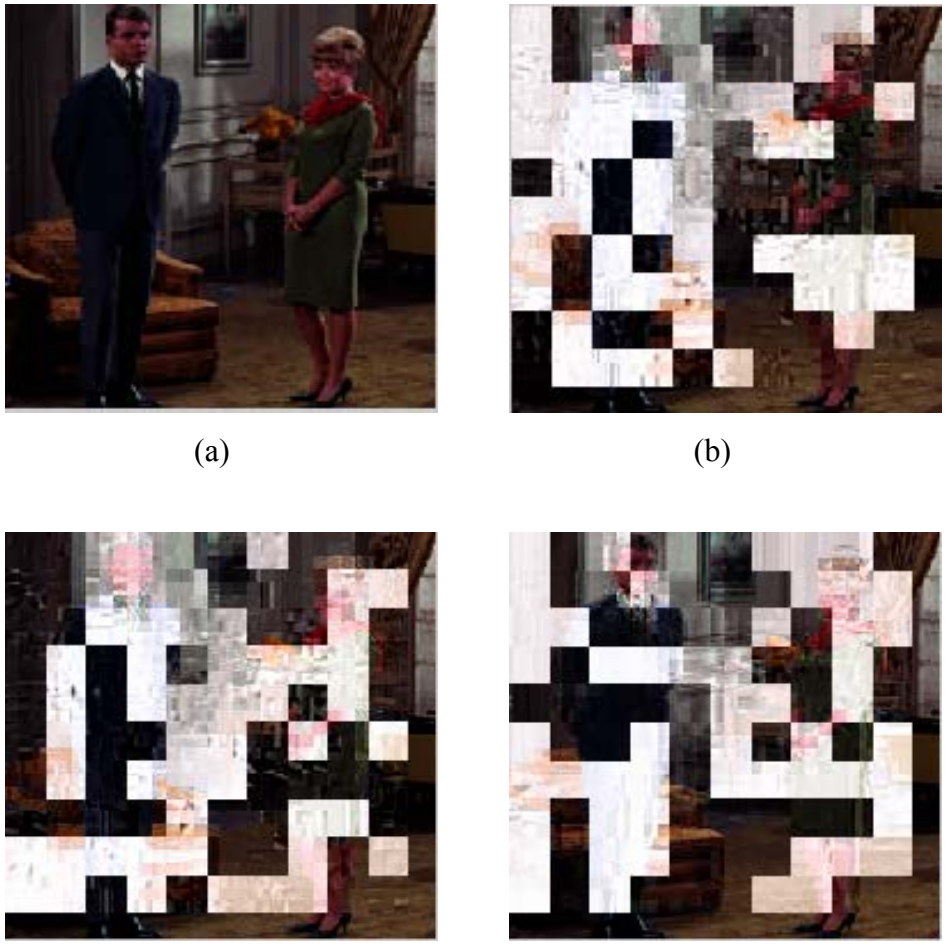


(a) (b)

(c) (d)

Figure 4.21 Key sensitivity test: (a) Original Sailboat 512x512 image, (b) encrypted compressed image using K1, (c) encrypted compressed image using K2, and (d) decrypted image in (b) using wrong key K2.

Their PSNR with respect to the original image is 10.6085 dB, and 13.2436 dB respectively. We can observe that in the selective encryption, no visual information can be recovered by replacing the ciphered DCT coefficients with constant values. So, our schema is not vulnerable to replacement attacks.

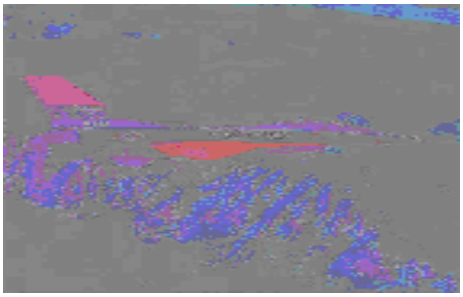


(a) (b)

(c) (d)

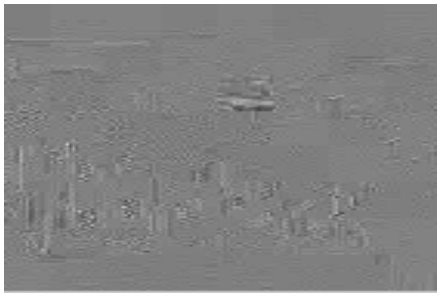
Figure 4.22 Key sensitivity test: (a) Original Couple 256x256 image, (b) encrypted image using K1, (c) encrypted image using K2, and (d) decrypted image in (b) using wrong key K2.

(a)



(b)

(c)



(d)

Figure 4.23 Attack in the selectively encrypted images by removing the encrypted data: (a) original 512x512 Airplane image, (b) decrypted image of the Airplane, (c) original 256x256 Goldhill image, and (d) decrypted image of the Goldhill.

### 4.6 Time Analysis

Experiments on various images show that the proposed encryption scheme is of high speed. Here, taking various images for example, the ratios between encryption/decryption process and encoding/decoding process are shown in Table 4.3. Where, the computer used in this test is Intel (R) Core ™ 2 Due CPU 2.8GH with 1.96 GB memory.

As may be seen from Table 4.3 the time ratio reduces with the rise of image size, since the compression process in the bigger image size is more complex and takes longer time than in the smaller images, so whatever increasing in the encryption time of the bigger images will be still smaller comparing to the compression time in them. For different images, the time ratios are all not higher than 17%, which means that the encryption process is of low cost, thus it may satisfy real-time requirement of many applications.

Table 4.3 Encryption time/compression time ratio test

| **Num. of blocks in segment** | **Image size** | **Encryption time/compression time ratio (%)** | |
|---|---|---|---|
| | | **Lowest Security** | **Highest Security** |
| **9** | 256x256 | 3.9552 | 16.5163 |
| | 512x512 | 2.72316 | 10.3459 |
| | 1024x1024 | 1.072 | 3.533 |
| **16** | 256x256 | 4.036 | 17.4323 |
| | 512x512 | 2.75995 | 10.492 |
| | 1024x1024 | 1.199 | 4.843 |

# Chapter 5: COMPARISON WITH OTHER RESEARCHES

In this chapter, the comparison between our scheme and other related works [22] are introduced based on specific evolution criteria.

## 5.1 COMPARISON CRITERIA

We need to define a set of evaluation criteria that will help evaluating and comparing selective encryption algorithms. Some criteria listed below are gathered from the literature.

### (I) Tunability (T)

Most of the proposed algorithms in the literature use fixed encrypted part. This property limits the usability of the algorithm to a restricted set of applications. It could be very desirable to be able to dynamically define the encrypted part with respect to different applications and requirements.

### (II) Visual degradation (VD)

This criterion measures the perceptual distortion of the cipher image with respect to the plain image. It assumes that the cipher image can be decoded and viewed without decryption. This assumption is not satisfied for all existing algorithms. In some applications, it could be desirable to achieve enough visual degradation, so that an attacker would still understand the content but prefer to pay to access the unencrypted content.

However, for sensitive data (e.g., military images/videos), high visual degradation could be desirable to completely disguise the visual content. Visual degradation is a subjective criterion that is why it is difficult to define a threshold for acceptable visual distortion regarding a given application

**(III) Encryption ratio (ER)**

This criterion measures the ratio between the size of the encrypted part and the whole data size [35, 36]. Encryption ratio has to be minimized by selective encryption.

**(IV) Cryptographic security (CS)**

Most of the research works on selective encryption evaluate the security level based only on visual degradation. In [37], Tang proposes a selective encryption algorithm based on DES encryption of DC coefficients and replacing the zigzag scan of the AC coefficients by a random permutation. The visual degradation achieved is very high, but the cryptographic security of the algorithm is very weak as pointed out in [38]. The cryptographic security should rely on:

(i) The encryption key (of a well-scrutinized encryption algorithm),

(ii) Unpredictability of the encrypted part.

**(V) Compression friendliness (CF)**

A selective encryption algorithm is considered compression friendly if it has no or very little impact on data compression efficiency. Some selective encryption algorithms impact data compressibility or introduce additional data that is necessary for decryption. It is desirable that this impact remains limited.

**(VI) Format compliance (FC)**

The encrypted bitstream should be compliant with the compressor. Any standard decoder should be able to decode the encrypted bitstream without decryption. This property is very important because it allows preserving some features of the compression algorithm used (e.g., scalability).

## 5.2 COMPARISON WITH OTHER WORKS

Podesser, Schmidt and Uhl, 2002. In [26], a selective bitplane encryption (using AES) is proposed, several experiments were conducted on 8-bit grayscale images, and the main results retained are the following: (1) encrypting only the MSB is not secure; a replacement attack is possible [26], (2) encrypting the first two MSBs gives hard visual degradation, and (3) encrypting three bitplanes gives very hard visual degradation. The comparesion criteria measurements will be as the following:

(a) Tunability: the algorithm is not tunable; a fixed bits need to be encrypted to guarantee confidentiality.

(b) Visual degradation: for 8 bits per pixel uncompressed image, hard visual degradation can be observed for a minimum of 3MSB bits encrypted.

(c) Cryptographic security: even when a secure cipher is used (AES), the selective encryption algorithm proposed is vulnerable to replacement attacks [26]. This attack does not break AES but replaces the encrypted data with an intelligible one. It is worth to note that visual distortion is a subjective criterion and does not allow to measure security as illustrated in this reference.

(d) Encryption ratio: at least 3 bitplanes over 8 (more than 37.5%) of the bitstream have to be encrypted using AES to achieve sufficient security.

(e) Compression friendliness: this algorithm is intended for uncompressed data. However, important bandwidth expansion is introduced by selectively encrypting MSBs which adversely impact the compressibility of encrypted images.

(f) Format compliance: it is format compliant.

(g) Data type: uncompressed image.

Zeng and Lei, 2003. In [39], selective encryption in the frequency domain (8×8 DCT and wavelet domains) is proposed. The general scheme consists of selective scrambling of coefficients. The 8×8 DCT coefficients can be considered as individual local frequency

components located at some subband. The scrambling operations (block rotation and sign changing) can be applied on these "subbands." I-, B-, and P-frames are processed in different manners. For I-frames, the image is first split into segments of macroblocks (e.g., a segment can be a slice), blocks/macroblocks of a segment can be spatially disjoint and chosen at random spatial positions within the frame. Within each segment, DCT coefficients are rotated. Then, sign bits of AC coefficients are randomly changed and DC coefficients (which are always positive for intracoded blocks) are flipped with respective threshold (e.g., 2558/2 = maximum DC value/2). There may be many intracoded blocks in P- and B-frames. Sign bits of motion vectors are also scrambled. The comparesion criteria measurements will be as the following:

(a) Tunability: not tunable.

(b) Visual degradation: high-visual degradation is achieved. Indeed, most of the image energy is concentrated in DC coefficients, thus, encrypting them affects considerably the image content.

(c) Cryptographic security: a large key space is obtained due to the use of equivalent Hadamard matrices in the scrambling. The Hadamard matrix-based encryption has insufficient diffusion, this leads to a reduction in key space. Experimental results show that when guessing 100 random keys, the best recovered image has low-visual degradation compared to the unencrypted one.

(d) Encryption ratio: variable, it depends on the number of coefficients to scramble.

(e) Compression friendliness: limited bandwidth expansion is allowed by this proposal. However, the major drawback of this scheme is that the encryption is lossy. Indeed, the encryption process implies a rounding operation that induces precision loss (so inadequate to lossless compression).

(f) Format compliance: it is format compliant.

(g) Data type: video.

Droogenbroeck and Benedett, 2002. The JPEG Huffman coder terminates runs of zeros with code words/symbols in order to approach the entropy. Appended bits are added to these codewords to fully specify the magnitudes and signs of nonzero coefficients, only these appended bits are encrypted (using DES or IDEA) [24]. The comparesion criteria measurements will be as the following:

(a) Tunability: not tunable.

(b) Visual degradation: high visual degradation is achievable.

(c) Cryptographic security: about 92% of the data is encrypted using well-scrutinized symmetric ciphers. It would be very difficult to break the encryption algorithm or try to predict the encrypted part.

(d) Encryption ratio: very high encryption ratio is required (about 92%).

(e) Compression friendliness: the encryption is separated from the Huffman coder and has no impact on the compression efficiency.

(f) Format compliance: JPEG compliant.

(g) Data type: image.

Cheng and Li, 2000. In [23], selective encryption is proposed for quadtree compression algorithm. The compressor output is partitioned into two parts; an “important part” that consists of the quadtree structure, and an “unimportant part” that consists of the leaf values. No encryption algorithm is specified, only the important part is encrypted. The comparesion criteria measurements will be as the following:

(a) Tunability: not tunable.

(b) Visual degradation: high visual degradation can be achieved only for images with high information (many colors, details, etc.).

(c) Cryptographic security: no encryption algorithm is specified.

(d) Encryption ratio: The encrypted part can reach about 50%.

(e) Compression friendliness: the encryption is performed after compression, no impact on the compression efficiency is observed.

(f) Format compliance: quadtree is not part of any compression standard.

(g) Data type: image.

Pommer and Uhl, 2003. The algorithm proposed in [25] is based on AES encryption of the header information of wavelet packet encoding of an image; this header specifies the subband tree structure. The comparesion criteria measurements will be as the following:

(a) Tunability: not tunable.

(b) Visual degradation: the encrypted content cannot be viewed without decryption.

(c) Cryptographic security: not secure against attack. Because statistical properties of wavelet coefficients are preserved by the encryption, then the approximation subband can be reconstructed.

(d) Encryption ratio: the encrypted part represents a very small fraction of the bitstream.

(e) Compression friendliness: the subband tree is pseudorandomly generated. This adversely impacts the compression efficiency.

(f) Format compliance: not format compliant; the encoder does not use standard wavelet packet decomposition.

(g) Data type: image.

Tang, 1996. The basic idea of the selective encryption algorithm proposed in [40] is to selectively encrypt I-frames of the MPEG stream; DES on DC coefficients and random permutation on the AC coefficients instead of the standard zigzag. This is done before compression. The comparesion criteria measurements will be as the following:

(a) Tunability: the algorithm is not tunable.

(b) Visual degradation: since intraframes are very important in MPEG compression (all B- and P-frames are computed according to I-frames), by encrypting them, high-visual degradation is achieved.

(c) Cryptographic security: the AC coefficients zigzag scan used in I-frames encoding is replaced by a pseudorandom permutation. Statistics of the AC coefficients are preserved. Therefore, the cipher images are feasible and allow recovering all AC coefficients.

(d) Encryption ratio: not specified.

(e) Compression friendliness: the nonoptimal scanning of the DCT coefficients introduces loss in compression efficiency. Indeed, this adversely affects Huffman encoding (due to distortion of the probability distribution of run-lengths for AC coefficients).

(f) Format compliance: the proposed scheme is compliant to JPEG and MPEG standards.

(g) Data type: image and video.

On the other hand, our proposed approach, selective encryption of images based on differential evolution, has the following properties:

(a) Tunability: the algorithm can be considered tunable since many security levels are allowed depending on the encryption parameters n and m, which can be fined-tuned to control visual distortion.

(b) Visual degradation: very high visual degradation can be observed from the results of Section 4.1, and the encrypted content cannot be viewed without decryption. The visual degradation can be fine-tuned using the encryption parameters (m, n), which change the number of coefficients to scramble. The encrypted content cannot be viewed without decryption.

(c) Cryptographic security: the algorithm can be considered as secure enough. Since cryptanalysis of selective encryption algorithms rely on key recovery (if encryption key space is not large enough) or prediction of encrypted part.

The generator of the key must not be subject to observation or manipulation by an adversary. So, our proposed schema uses cryptographically secure pseudo-random number generator (PRNG) to generate key space size over than $12\times10^{23}$ which is large enough to make brute-force attack infeasible. In addition, many security levels can be obtained by changing the number of blocks and segments of DCT coefficients needed to encrypt. Replacing the encrypted DCT coefficients with a fixed value still gives an intelligible version of the image as shown in the Section 4.6; hence, our selective encryption algorithm is not vulnerable to replacement attacks. Also, it is not vulnerable to statistical attacks as can be seen from on the results illustrated in Section 4.2.

(d) Encryption ratio: in the crossover operation, 14 AC coefficients of luminance block in the segment are crossed with the same frequency coefficients of another block in a different segment, so the ratio between encrypted pixels and image pixels is 14 / (3x64), where each block consists of 64 pixels, and no chrominance planes are used in the encryption.

In the scale operation, the DC coefficient of luminance block is scaled with constant, so the ratio between encrypted pixels and image pixels is 1/ (3×64).

In the mutation operation, the sign bit of the first 30 DCT coefficients of the block are mutated according to the mutation function, while the number of the blocks to mutate, and the number of the segment are determined by the user, so the ratio between encrypted bits and image bits is $\{30 \times blocknum \times (segnum / L)\} / \{H \times W \times pixelbits\}$. Where the number of the block in each segment is ranged from 1 to 9, if segment includes 9 blocks, or from 1 to 16, if the segment consists of 16 blocks. While number of segments segnum $= H \times W / (segblock \times 64)$, where $H$ and $W$ are the height and the width of the image , $segblock$ is 9 or 16, and $L$ is the user parameter which determines the number of the segments used in the mutation, and ranges from 2 to 5.

Thus, in the cryptosystem, the ratio between encrypted pixels and image pixels is:

$$R= \frac{14}{(3\times 64)}+\frac{1}{(3\times 64)}+\frac{30\times(blocknum)\times\left(\frac{Segnum}{L}\right)}{H\times W\times pixelbits} \tag{5.1}$$

$$R= \frac{15}{192}+\frac{30\times blocknum}{64\times L\times segblock\times pixelbits} \tag{5.2}$$

It is apparent from Equation 5.2 that the R is in inverse proportion to *L*, *segblock*, and *pixelbits*, and in proportion to blocknum. Taking *segblock* = 16 for example, and each pixel = 24 bits (for color image), then R = 0.087 for high security level (where *L* = 2 and *blocknum* = 16), otherwise R = 0.0783 for low security level (where *L* = 5 and *blocknum* = 1); and if each pixel = 8 (gray scale image), then R = 0.1074 for high security level (where *L* = 2 and *blocknum* = 16), otherwise R = 0.0788 for low security level (where *L* = 5 and *blocknum* = 1); and so on as shown in Table 5.1 . As may be seen that, the encryption data ratio R is relatively small, which ensures that the real-time requirement may be satisfied.

Table 5.1 Encryption ratio R, for different segment size

| **Segment size (block)** | **Pixel size (bit)** | **Low security (%)** | **High security (%)** |
|---|---|---|---|
| **9** | 8 | 7.94 | 10.74 |
| | 24 | 7.85 | 8.78 |
| **16** | 8 | 7.88 | 10.74 |
| | 24 | 7.83 | 8.78 |

(e) Compression friendliness

The whole encryption scheme affects compression ratio slightly. In the proposed encryption scheme, sign mutation of DCT coefficients keeps compression ratio unchanged according to JPEG encoding. The scale method affects compression ratio

slightly, because the adjacent relationship among blocks is kept, and only one DC coefficient in each block is scaled referencing to others.
Subband crossing of DCT coefficients of the same frequency level avoids the crossing between bigger ones and smaller ones, which has a slighter effect on compression ratio than global crossover method does.

JPEG encoding (loss mode) realizes bit-rate conversion by changing quantization steps. The encryption scheme proposed here changes only the signs of DCT coefficients and the relative position of DCT coefficients, but leaves the ranges of DCT coefficients unchanged. Therefore, the encryption scheme supports compression efficiency. That is to say, it permits to decompress the compressed and encrypted data before decrypting it firstly.

The encrypted images can be decrypted correctly although they have been directly compressed with different compression ratios. The experimental results of Airplane and Couple images are shown in Figure 5.1. Images (b)-(f) are decrypted from the ones that are compressed with compression ratio varying from 10% to 80%. It is apparent that the images are all decrypted correctly with little impact on compression efficiency (less than 7%), which shows that the encryption scheme proposed here supports direct bit-rate control, and it is compression-friendliness.

(e) Format compliance: since the encryption happens in the frequency domain prior to entropy coding and bitstream formation, the resulting encryption compressed bitstream will fully "conform" to the JPEG compression standard, i.e., a conventional decoder will be able to decode the encryption compressed bitstream as if it were an ordinary compressed bitstream.

(g) Data type: image.

Table 5.2 summarizes the related work with respect to each criteria described above. The main symbols used are
(1) "+" for satisfied criterion,
(2) "-" for nonsatisfied criterion,

(3) “H” for high,

(4) “?” for nonspecified,

(5) “T” for tunability,

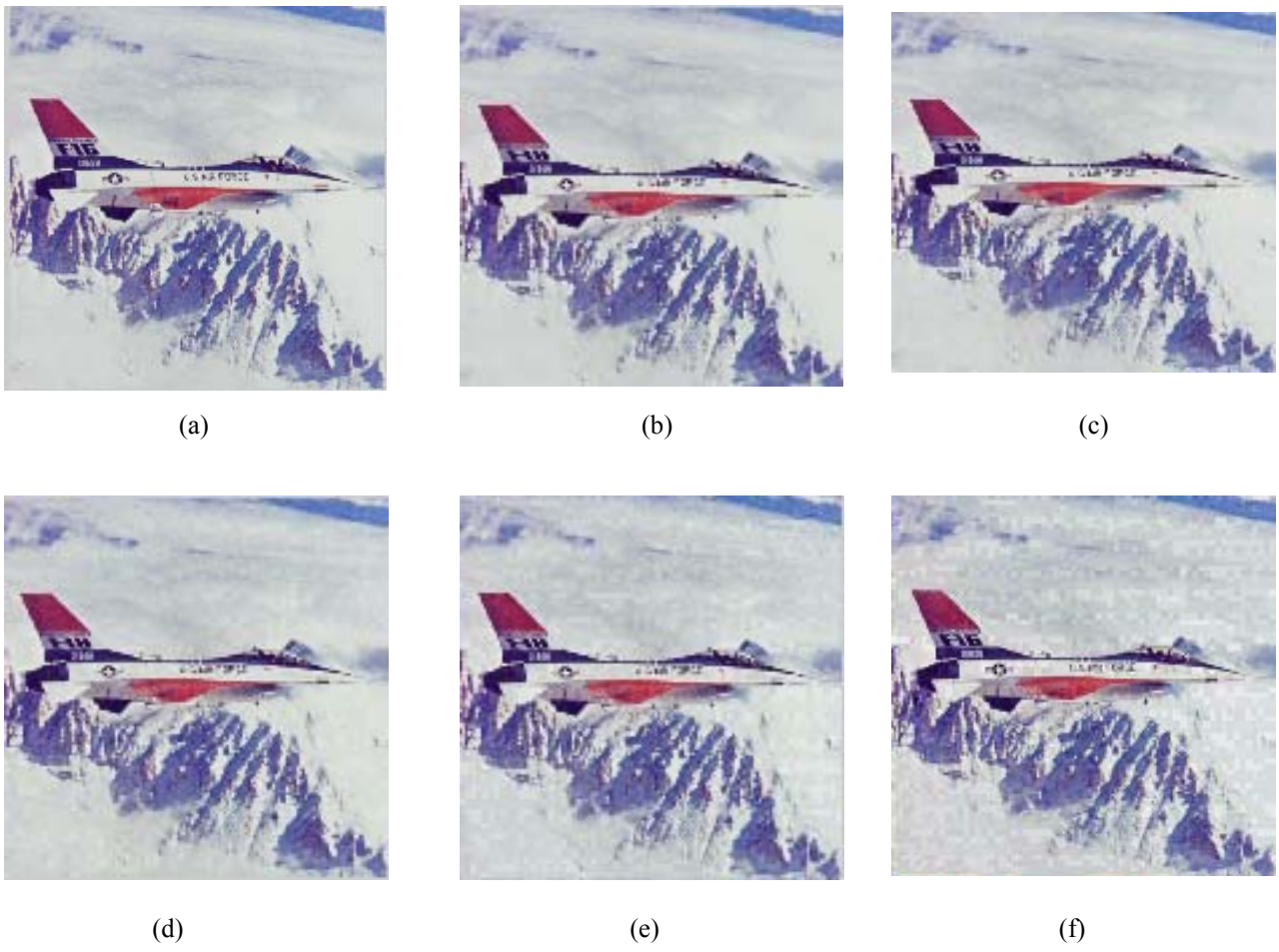
(6)”VD” for visual degradation,

(7) “CS” for cryptographic security,

(8)”ER” for encryption ratio,

(9)”CF” for compression friendliness,

(10)”FC” for format compliance.



(a) (b) (c)

(d) (e) (f)

Figure 5.1 Decrypted airplane image after bit-rate control: (a) original image, (b) compression ratio = 10%, PSNR = 38.9767 dB and compression lost = 1.03%, (c) compression ratio = 30%, PSNR = 34.7997 dB and compression lost = 2.79%, (d) compression ratio = 50%, PSNR = 32.6004 dB and compression lost = 4.46%, (e) compression ratio = 70%, PSNR = 29.9254 dB and compression lost = 5.55%, and (f) Compression ratio = 80%, PSNR = 27.4129dB and compression lost = 5.84%.

It is desirable that visual degradation is variable and dynamically tunable to adapt to different application requirements. Encryption ratio needs to be minimized. Grayed boxes indicate unsatisfied criteria.

Table 5.2 Summary of related work with respect to each criterion

| Related Work | T | VD | CS | ER | CF | FC |
|---|---|---|---|---|---|---|
| **Tang [40],1996** | **-** | **H** | **-** | **?** | **- (compressiondrop=40%)** | + |
| **Podesser, Schmidt , Uhl [26] , 2002** | **-** | **H** | **-** | **-(>37.5%)** | **-** | + |
| **Zeng and Lei [39], 2003** | **-** | **H** | **-** | **- 20%** | + **(compression drop<5%)** | + |
| **Pommer and Uhl [25] , 2003.** | **-** | **-** | **-** | + | **-** | **-** |
| **Cheng and Li [23] ,2000** | **-** | + | **-** | **50%** | + | **-** |
| **Droogenbroeck , Benedett [24],2002** | **-** | **H** | + | **- 92%** | + | + |
| **Proposed algorithm** | + | **H** | + | + **( < 10.8 %)** | + **( compression drop<6%)** | + |

# CHAPTER 6: CONCLUSION AND FUTURE WORK

## 6.1 CONCLUSION

In this research, selective encryption scheme of images based on Differential Evolution operations (crossover and mutation) is proposed.

The DCT image data are encrypted efficiently by employing selective crossover of DCT coefficients of the same frequency bands between selective segments. In addition, mutation function modifies the sign bit of certain DCT coefficients restricted to a specific number of blocks and segments determined by the user needs. Finally, scaling of DC coefficients is done to diffuse statistics.

The proposed encryption techniques appear to achieve a very good compromise between several desirable properties such as tunability, visual degradation, compression friendliness, security, format compliance, and encryption ratio; therefore is very suitable for network image applications. In short, we can say that the proposed scheme:

- It is a flexible selective encryption scheme that is tunable and allows to trade of a certain number of parameters to target different applications with different requirements in terms of security or visual degradation.
- It is ensured that the real-time requirement may be satisfied by keeping the encryption ratio less than 11%.
- It allows preserving some features of the compression algorithm used (e.g., scalability), since It achieves format compliance where the decoder can be able to decode the encrypted bitstream without decryption.
- It has little impact on compression efficiency.
- It resists the statistical and replacement attacks with high sensitivity of the key generated based on LFSR.

## 6.2 FUTURE WORK

In future works, we will focus on designing selective encryption algorithms for any compression algorithm. We believe that some compression algorithms are more cooperative and could be better candidates for selective encryption. At least, we will develop our proposed scheme for JPEG2000 [11], the wavelet transform based image coding system. Its characteristics (embedded encoding, block-based encryption, many progression orders, local region access, etc.) may help in designing flexible scheme, in order to meet a larger set of requirements and target more applications.

Development of a general selective encryption scheme for still images and video with the listed desirable objectives is a complex problem. Ideally, our findings for images should be extendable to video protection with more computational requirements. Identification of the most significant parts of multimedia data is a key step for selective encryption.

The proposed algorithm can be extended to use multiple encryption concept where the image has multiple owners and each of them encrypts some DCT coefficients using his private key. In addition, another extension can be applied by using over-encryption concept where encrypted DCT coefficients using the key K1 are decrypted using the new Key K2, then will be encrypted again with the original key K1. All of these extensions have better performance and higher security level.

**REFERENCES**

[1] A. M. Eskicioglu, J. Townand, and E. J. Delp, "Security of Digital Entertainment Content from Creation to Consumption," S*ign. Procs.: Image Comm.,* vol. 18, no. 4, pp. 237-262, April 2003.

[2] B. Schneier, "Encryption Of Images," *in Applied Cryptography*, 2nd ed. New York: Wiley, 1996, pp. 20-70.

[3] T. Lookabaugh, "Selective Encryption, Information Theory, and Compression," *in 38th ASILOMAR Conference on Signals, Systems and Computers*, vol. 1, California, USA, 2004, pp. 373–376.

[4] X. Liu and A.M. Eskicioglu, "Selective Encryption of Multimedia Content in Distribution Networks: Challenges and New Directions," *in IASTED International Conference on Communications, Internet and Information Technology,* Scottsdale, AZ, Nov. 2003, pp.527–533.

[5] D. Karabo and S. Okdem, "A Simple and Global Optimization Algorithm for Engineering Problems: Differential Evolution Algorithm," *Turk. J. Elec. Engin.,* vol.12, no.5, pp. 125-134, 2004.

[6] R. Storn, and Kenneth Price, "Differential Evolution - A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces," Elect. Eng. Res. Lab., Berkeley, Tec. Rep. TR-95 (2312- 25)-3, 1995.

[7] R. C. Gonzalez and R. E. Woods, "Compressions Standards," in *Digital Image Processing*, 3rd ed. Elsevier: Pearson Education, 2002.

[8] A. Léger and T.Omachi, "The JPEG Still Picture Compression Algorithm," *Opt. Engin.*, vol. 30, no. 7, pp. 947-954, July 1991.

[9] A.Léger, M. Mitchell, and Y. Yamazaki, "Still Picture Compression Algorithms Evaluated for International Standardization," *IEEE Comm. Soct.*, vol. 10, no. 8, pp. 228-235, Nov. 1988.

[10] W. Pennebaker and J. Mitchell, “JPEG: Still Image Data Compression Standard,” *Van Nostrand Reinhold*, vol. 53, no. 14, p. 475-487, 1993.

[11] T. Acharya and P. Tsai, “Source Coding Algorithms,” *in 2000 Standard for Image Compression Concepts, Algorithms*, 3rd ed. New York: Wiley, 2005, pp. 23-53.

[12] Xiliang Liu, “Selective Encryption of Multimedia Content: Challenges and New Directions,” *Proc. of Comm*., *Inter., and Inform. Tech.,* vol. 41, pp. 492-503, Nov. 2003.

[13] W. Zeng and S. Lei, “Efficient Frequency Domain Selective Scrambling of Digital Video,” *IEEE Trans. on Mult*., vol. 5, no. 12, pp. 118–129, 2003.

[14] K.R. Rao, and P. Yip, “Discrete Cosine Transform-Algorithms, Advantages, Applications,” *Proc. of Comm*., vol. 46, no. 4, pp. 118–129, 1990.

[15] M. Barni, “The JPEG Family of Coding Standards,” *in Document and Image Compression*, 2nd ed. France: Taylor and Francis Group, 2004, pp. 87-113.

[16] P.G. Howard, and J.S. Vitter, “New Methods for Lossless Image Compression Using Arithmetic Coding,” Columbia University Dept. of Computer Science, New York, Tech. Rep.CS-91-47, 1991.

[17] C. Wu and C. Kuo, “Design of Integrated Multimedia Compression and Encryption Systems,” *IEEE Trans. on Mult.*, vol. 7, no. 5, pp. 828–839, Oct. 2005.

[18] A. Massoudi, F. Lefebvre, C. De Vleeschouwer, B. Macq, and J. Quisquater, “Overview on Selective Encryption of Image and Video: Challenges and Perspectives,” *Hindawi Pub. Corp. EURASIP Journal on Inf. Sec.,* vol. 2008, no. 179290, pp. 42-60, 2008.

[19] W. Puech and J. Rodrigues, “Crypto-Compression of Medical Images by Selective Encryption of DCT,” *in 13th European Signal Processing Conference*, Turkey, Sep. 2005, pp. 235-242.

[20] C. E. Shannon, "Communication Theory of Secrecy Systems," *Bell Syst.Tech. Journal*, vol. 28, no. 5, pp. 656-715, 1949.

[21] A. Said, "Measuring the Strength of Partial Encryption Scheme," in ICIP 2005, *IEEE International Conference in Image Processing*, vol. 2, Genova, Italy, 2005, pp. 1126–1129.

[22] A. Massoudi, F. Lefebvre, C. De Vleeschouwer, B. Macq, and J.-J., Quisquater, "Overview on Selective Encryption of Image and Video: Challenges and Perspectives," *Hindawi Pub. Corp. EURASIP Journal on Inf. Sec.*, vol. 45,no. 2, pp. 1329–1335, Nov. 2008.

[23] H. Cheng and X. Li, "Partial Encryption of Compressed Images and Videos," *IEEE Trans. on Sign. Proc.,* vol. 48, no. 8, pp. 2439–2451, 2000.

[24] M. Van Droogenbroeck and R. Benedett, "Techniques for a Selective Encryption of Uncompressed and Compressed Images," *in Proceedings of Advanced Concepts for Intelligent Vision Systems Conf.*, Ghent, Belgium, Sep. 2002, pp. 90–97.

[25] A. Pommer, and A. Uhl, "Selective Encryption of Wavelet-Packet Encoded Image Data," *ACM Mult. Sys.*, vol. 5, no. 4, pp. 111–119, 2003.

[26] M. Podesser, H. Schmidt, and A. Uhl, "Selective Bitplane Encryption for Secure Transmission of Image Data in Mobile Environments," *in 5th Nordic Signal Processing Symposium*, on board Hurtigruten, Norway, Oct. 2002, p. 45-56.

[27] M. B. I. Reaz, F. Mohd-Yasin, S. L. Tan, and H. Y. Tan, "Partial Encryption of Compressed Images: A Hardware Approach," *in 3rd International Conference: Sciences of Electronic, Technologies of Information and Telecommunications*, Tunisia, March 2005, pp. 70-76.

[28] K. Lim and J. Beom, "Improved on Image Transform Coding by Reduation Interblock Correlation," *IEE Transt. on Sig. Proc.,* vol. 4, no. 8, pp. 125-134, Aug. 1995.

[29] P. Wong, C. Oscar, W. C. Justy, ”Data Hiding and Watermarking in JPEG Compressed Domain by DC Coefficient Modification,” *SPIE Sym. of Sec. and Water. of Mult. Cont.* vol. 3971, no.4, pp.237-244, Sep. 2000.

[30] B. Y. Mohammad Ali and J. Aman, ”Image Encryption Using Block-Based Transformation Algorithm,” *IAENG Int. Comp. Senc*., vol. 35, no. 1, pp. 15-23, 2008.

[31] A. Mitra, Y V. Subba Rao, and S. R. M. Prasnna, ”A New Image Encryption Approach Using Combinational Permutation Techniques,” *Comp. Senc. of Multi.,* vol. 1, no. 1, p.127, 2006.

[32] W. Puech, J.M. Rodrigues, ”Analysis and Cryptanalysis of a Selective Encryption Method for JPEG Images,” *in Eighth International Workshop on Image Analysis for Multimedia Interactive Services, Santorini island, Greece*, June 2007, pp. 78-80.

[33] Y.B. Mao, G. Chen, S.G. Lian, “A Novel Fast Image Encryption Scheme Based on the 3D Chaotic Baker Map,” *Int. J. Bifurcat Chaos*, vol. 14,no. 21, pp. 3613–3624, 2004.

[34] M. Abd El-Wahed, S. Mesbah, and A. Shoukry, “Efficiency and Security of Some Image Encryption Algorithms,” *World Conference on Engineering,* vol. 1, London, U.K., 2008.

[35] W. Zeng and S. Lei, “Efficient Frequency Domain Selective Scrambling of Digital Video,” *IEEE Trans. on Mult*., vol. 5, no. 1, pp. 118–129, 2003.

[36] C. Wu, P. Kuo, “Efficient Multimedia Encryption via Entropy Codec Design,” i*n 3 rd Conference of Proceedings of SPIE Security and Watermarking of Multimedia Content,* vol. 43, San Jose, CA, Jan. 2001, pp. 802-814.

[37] L. Tang, “Methods for Encrypting and Decrypting MPEG Video Data Efficiently,” in *Proceedings of the 4th ACM International Multimedia Conference and Exhibition*, Boston, Mass, USA, Nov. 1996, pp. 219–229.

[38] L. Qiao, K. Nahrstedt, and M.-C. Tam, "Is MPEG Encryption by Using Random List Instead of Zigzag Order Secure?" *in Proceedings of the IEEE International Symposium on Consumer Electronics*, Singapore, Dec. 1997, pp. 226–229.

[39] W. Zeng and S. Lei, "Efficient Frequency Domain Selective Scrambling Of Digital Video," *IEEE Trans. on Mult., vol. 5,* no. 1, pp. 118–129, July 2006.

[40] L. Tang., "Methods for Encrypting and Decrypting MPEG Video Data Efficiently," *Proc. ACM Mult*, vol. 3, no. 5, pp. 219–229, 1996.

[41] J.M. Rodrigues W. Puech, and A.G Bors, "Selective Encryption of Human Skin in JPEG Images," *in IEEE International Conference on Image Processing*, Atlanta, GA, USA, 2006, pp. 1981–1984.

[42] M. Van Droogenbroeck, "Partial Encryption of Images for Real-time Applications," *Fourth IEEE Benelux Signal Processing Conference*, Hilvarenbeek, the Netherlands, 2004, pp. 11-15.