

Zagazig University
Faculty of computers and information
Computer science department

Secure and Authenticated Data Transmission

Graduation Project

Project group:

1. Hadir Abd_allah Ali.
(eng_reabsh@yahoo.com)
2. Hala Hassan Mohammed.
(eng.hala.etman@gmail.com)
3. Maha Abd_elhafez Abd_elhamed.
4. Maha Ismail Masoud.
(Eng_mohacs@yahoo.com)
5. Manar Hamdy Eltegany.
6. Nayera Mohamed Hassan.
(justsmile_2020@yahoo.com)

Project supervisor:

Dr/ Ahmed Rafat Abas

Faculty of Computers & Information – Zagazig University

FACULTY OF COMPUTERS & INFORMATION
ZAGAZIG UNIVERSITY – EGYPT
2010

Contents Page

Chapter 1: Introduction:

1.1 Problem Definition	5
1.2 Solution Overview	6
1.3 Document outline	7

Chapter 2: Color Modes and Media Formates:

2.1 Introduction.....	10
2.2 Color Models.....	14
2.2.1 RGBColorModel.....	14
2.2.2 CMY Color Model.....	15
2.2.3 YUVColor Model (also called Y, Cb, and Cr).....	16
2.3 Standards of some basic media formats.....	18
2.3.1 Bitmap(BMP)standard Format.....	18
2.3.2 JPEG standard format.....	20
2.3.3 AVI standard format.....	23

Chapter 3: Discrete Cosine Transform:

3.1 Introduction.....	26
3.2 The Discrete Cosine Transform.....	27
3.2.1 DCT.....	27
3.2.2 The One Dimensional Discrete Cosine Transform.....	28
3.2.3 The Two-Dimensional DCT.....	28
3.3 Properties of DCT.....	29
3.4 Forward Discrete Cosine Transfer (DCT).....	29
3.4.1 Features.....	29
3.4.2 Description.....	30

3.5 Forward DCT.....	30
3.6 Fast Discrete Cosine algorithm.....	32

Chapter 4: Authentication And Digital Signature:

4.1 Introduction.....	34
4.2 Authentication.....	35
4.2.1: Two-factor authentication.....	36
4.2.2: Four Levels of Proof.....	37
4.3: digital signature.....	37

Chapter 5: Design and implementation :

5.1 Introduction.....	48
5.2 System Design.....	49
5.2.1 Data Hiding System Design.....	50
5.3 System Implementation.....	51
5.3.1 Data Hiding System Implementation.....	51
5.3.2 Data Extracting System Implementation.....	54
5.4 User Interface.....	59

CHAPTER 1

INTRODUCTION

Chapter 1

Introduction

1.1 Problem Definition

The main problem in the age of the internet, speed and hackers is to be protected against attacks, sudden crashes or accidental destruction.

Your important and private data must be secured enough through transmissions. Securing data means that the data must be protected against any type of attacks, thus we need the science of cryptography that is concerned with securing data. The aim of cryptography has always been to provide secure communications over an insecure channel, such that only a set of intended recipients can understand the message. Cryptography uses various ranges of sciences, including mathematics, computer science, information theory and human psychology.

Any cryptographic systems must provide a number of different types of services. These services are the core, and to implement these services there is a need for mechanisms[1], which are briefly mentioned as follows:

Confidentiality which is the protection of the transmitted data from passive attacks "spy's", which be achieved by encryption algorithms.

Authentication, which is Assurance that the parties involved in a real-time transaction, are who they say they are, which be achieved by public private system.

Data Integrity which is assurance the data has not been modified by any unauthorized parties, which be achieved by digital signature.

Access Control which is the ability to limit and control the access to the host, which be achieved by key management system.

Non-repudiation is to prevent either sender or receiver from denying a transmitted message, which be achieved by Trusted Third Party "TTP".

Any attack threatens any service of last services and that attack make this service unavailable. The attacker may be someone strange or a member.

The power of the encryption algorithm, which is referred to the amount of secrecy, is measured in terms of computational efforts needed to break the system by deducing the key from the intercepted ciphertext. The problem of key extraction is the cryptanalysis. Cryptanalysis is a factor of time and computational power.

As the more secure encryption algorithms appear with longer keys and complicated mathematics the more computational power becomes doubled and the more algorithms and techniques be proposed by cryptanalysis scientists , really there is a very fast race.

1.2 Solution Overview:

In the race between cryptography and cryptanalysis, you have to save your data by more than one mean. The

main idea for not attracting the attention of the cryptanalysis is to hide data in a container that he would not expect that there is an important thing in this container.

Nowadays multimedia has been spread over the world. The container can be a multimedia file even image or sound or video. The multimedia files have a great interest of this domain, which called Data Hiding, because these files have a relative big size and there is no doubt on them and you can put a reasonable relative large size of data on it without sensible quality changed or be recognized.

In this domain, the file formats must be studied with all encoding, decoding and conversions concerned with this format. In the domain of multimedia, there is a lossy compression when converting from one format to another format. Thus, there is a need for error correction algorithm and an effective one because the errors may be nearby others.

Therefore, Data Hiding seems to be the solution for the problem of cryptography. However, if any one knows the way by which the data is saved in this file, he can extract data easily. That is the reason that data hiding in container is not enough. The data can be hidden in specified positions in the image which is not known by the attacker or the data can be encrypted and packed into the container file.

In a single network, there is a need for authenticating users. Thus, we must implement a method for determining the required information about the sender.

1.3 Document Outline:

This document provides the background material, theory and implementation issues for a data-hiding

algorithm, Forward Discrete Cosin Transformation, an error correction technique, finally an authentication system.

Chapter 2 aims to define the scope of the fields that we call multimedia and image processing, discuss briefly the principle approaches used in this field and define the major color systems used in media and how to transform a media from one to another.

The chapter also defines the major representations that an image or video can be represented with, typically, image representation in spatial domain and in frequency domain.

Here there is an overview of basic color models like RGB color model, CMY color model and YUV color model.

Chapter 3 aims to define the Forward Discrete Cosin Transform. Or DCT transforms data into a format that can be easily compressed. The characteristics of the DCT make it ideally suited for image compression algorithms. These algorithms let you minimize the amount of data needed to recreate a digitized image.

The purpose of the DCT transform is that instead of processing the original samples, you work with the frequencies present in the original image(frequency domain).

Chapter 4 this chapter will explain authentication concept. And how we can apply this concept in our project to be accurate who is the sender of the hidden data’

Chapter 5 This chapter will explain in details the design and implementation of the system. This includes the design of the over all system, the design of the main components of the system supported by block diagrams. The implementation of the system will also be introduced and illustrated by means of flow charts and pseudo code. We will support this by introducing snapshots taken from the real

system. The system integration will show how the system components will be integrated together to construct the over all system.

CHAPTER 2

COLOR MODELS AND MEDIA FORMATS

Chapter 2

Color Models and Media Formats

2.1-Introduction:

Interest in multimedia and digital image processing methods emerges from two principles application areas: improvement of pictorial information for human interpretation; and processing of image data for storage, transmission, compression, and representation for autonomous machine perception.

This chapter aims to define the scope of the fields that we call multimedia and image processing, discuss briefly the principle approaches used in this field and define the major color systems used in media and how to transform a media from one to another.

The chapter also defines the major representations that an image or video can be represented with, typically, image representation in spatial domain and in spectrum domain.

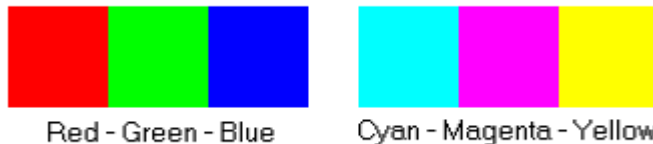
The chapter gives some brief description about compression techniques both lossy and lossless and highlights the standard formats of images and video and how to encode some media in one format and how to decode it.

What is a Color Model?

Color: is a sensation produced by the human eye and nervous system. It is related to light, but an understanding of the properties of light is not sufficient to understand color, and is especially not sufficient to understand the art of color reproduction. Overwhelming experimental evidence tells us that the perception of a color is related to the strength of three signals which are transmitted along the optic nerve to the brain. The importance of this is that:

It is useful to represent a color by a set of exactly three numbers.

In practice, the set of three numbers must be related to some actual color reproduction process. The numbers commonly specify portions of some set of **primary colors** such as:



As everyone learned in grade school, mixing varying portions of primary colors enables one to perform color reproduction. There are **two big problems** with using numbers that correspond to these primary colors to communicate about color.

My magenta may be a little different from your magenta.

The specification 20% Cyan, 60% Magenta, 47% Yellow fully specifies a color only if everyone uses inks that are of the same color. Offset printers have achieved this (to some degree) for many years by insisting that everyone use the same "process inks".

Too bad the makers of digital printers couldn't do the same, but hey - process inks are pretty restrictive anyway.

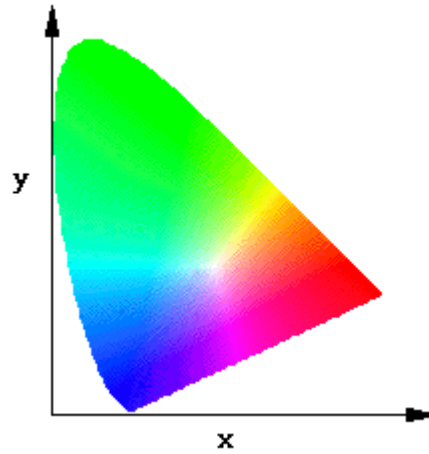
A little shift in yellow makes a bigger difference in color than a little shift in magenta.

In the above mentioned color; 20% Cyan, 60% Magenta, 47% Yellow, a change of a few percent in Yellow will have a **much** bigger affect on the *perceived* color than a similar change in Cyan. This is true in general, and makes it very tough for a computer program to figure out what is and isn't important about a color specification. Color scientists say that the color model (space) is not "perceptually uniform".

It would be very useful, especially when computers need to determine the differences between colors, if we could measure colors in such a way that the "distance" between two sets of numbers corresponded to the perceived difference between the colors.

A computer could then make better decisions about the "goodness" of color reproduction.

Fortunately, such a "scientific" system of color measurement was devised way back in the 1930's, the CIE color system. These numbers can be measured by instruments called "**colorimeters**" (as opposed to "**densitometers**"). The famous CIE "chromaticity diagram" illustrates some relationships between colors in this "color space" and is illustrated below.



The real magic of CIE numbers goes beyond the "perceptual uniformity" issue. An even bigger advantage is that this was defined decades ago and nobody owns it so everybody can agree on it. Makers of scanners and makers of printers, even the most entrenched competitors, can all form some agreement on what is meant by a particular set of CIE color numbers.

Considering the situation of the last ten years, that is real magic!

A "**color model**":- is any method of associating names or numbers with colors. Examples include CMYK dot densities as reproduced by an offset press, RGB voltages sent to the electron guns of a cathode ray tube display, CIE LAB numbers as measured by a "colorimeter", or Pantone color numbers as seen on the "swatch books" published by that company.

The importance of a color model is that it allows communication about what color should be produced. When a graphic designer calls a printer on the telephone to specify a color by referring to a swatch book, he is doing this. When two computer programs communicate through a digital CIE LAB specification, they're doing much the same thing.

Some color models allow us to determine very effectively whether two colors are "similar" just by examining these numbers. This is very useful when we want a computer to be able to decide if two colors are "similar". In the case of the Pantone numbers, where numbers do nothing other than *name* the colors, this is scarcely possible at all. "Device" color models such as RGB and CMYK, which use numbers that are very directly related to the physical color reproduction process are better. "Scientifically designed" color models such as CIE LAB are better still, making them preferred for computer calculation of color reproduction problems.

When it is possible to define a "distance" between two sets of numbers that bears some relation to the difference between two colors, the color model may be referred to as a **color space**

2.2 Color Models:[2]

2.2.1 RGB Color Model:

- A color image is represented in an RGB color model in a 2-D array of (R,G,B) integer triplets ,each pixel has its three values. These triplets will represent the corresponding color for their pixel (see figure 2.1).
- RGB color model used in some image formats like BMP image format also the model used in displaying devices , for each pixel has RGB triplet, these triplets encode how much the corresponding phosphor should be excited in devices such as a monitor.
- The main advantage for RGB model is simplicity and the main disadvantage for RGB model is when it is implemented it has great size files (e.g. Bitmap versus JPEG).

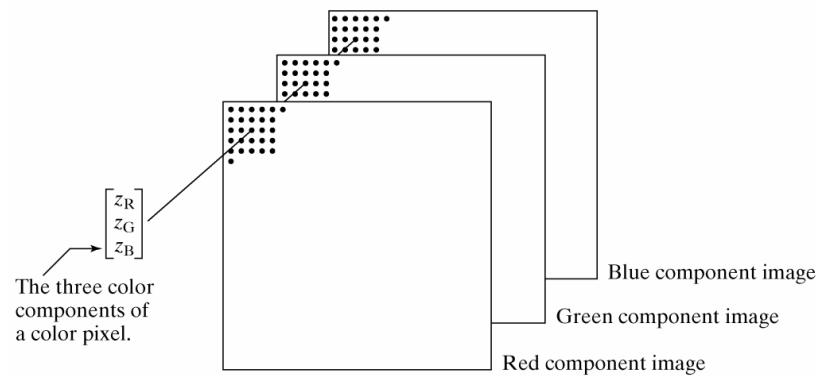
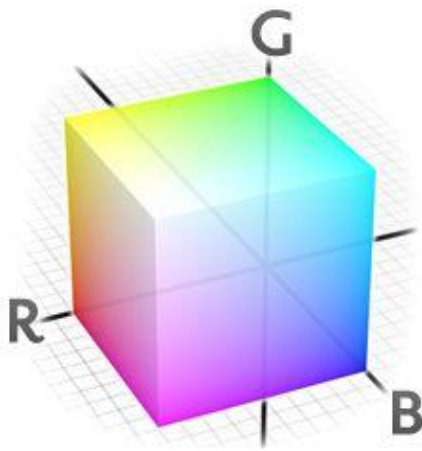


Figure 2.1 RGB components.[2]



RGB uses additive color mixing that produces secondary colors where two colors overlap, while equal intensities of all three colors produce white

2.2.2 CMY Color Model:

- Cyan, Magenta, and Yellow (CMY) are complementary colors of RGB. They can be used as subtractive primaries.
- CMY model is mostly used in printing devices where the color pigments on the paper absorb certain colors (e.g., no red light reflected from cyan ink).

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2.1)$$

Relationship between RGB and CMY is illustrated in figure 2.2.

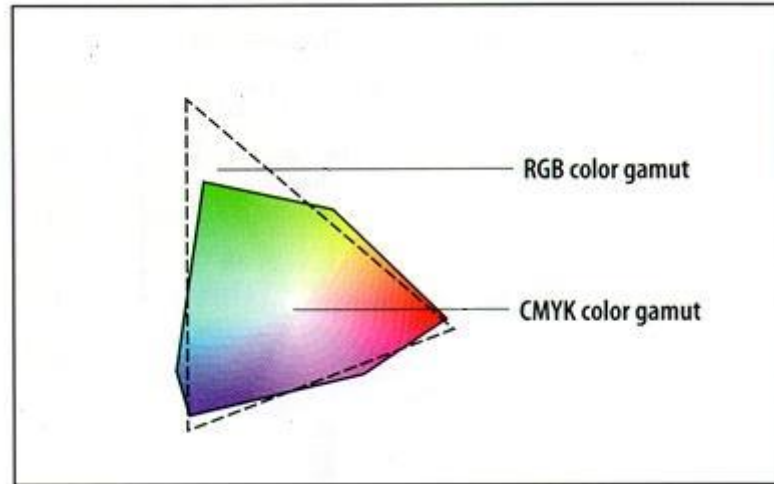


Figure 2.2 Relationship between CMY and RGB color systems.

- The RGB gamut is smaller, hence certain visible colors (e.g. pure yellow, pure cyan) cannot be seen on monitors
- The CMYK gamut is the smallest (but not a straight subset of the RGB gamut).

2.2.3 YUV Color Model (also called Y, Cb, and Cr):

- The YUV color model is such an important model, this model considers the human retina mechanism. The eye, particularly the retina, has two kind of cells as visual analyzers :
 - Cells for night view which perceive only nuances of gray ranging from intense white to the darkest black,

and cells for the day view which perceive the color nuance.

- Given an RGB color, the first cells detect a gray level similar to that given by the luminance value. This is represented by Y component.
- The second cells, responsible for the perception of the color nuance, are the cells which detects a value related to that of the chrominance. This is represented by U and V components.

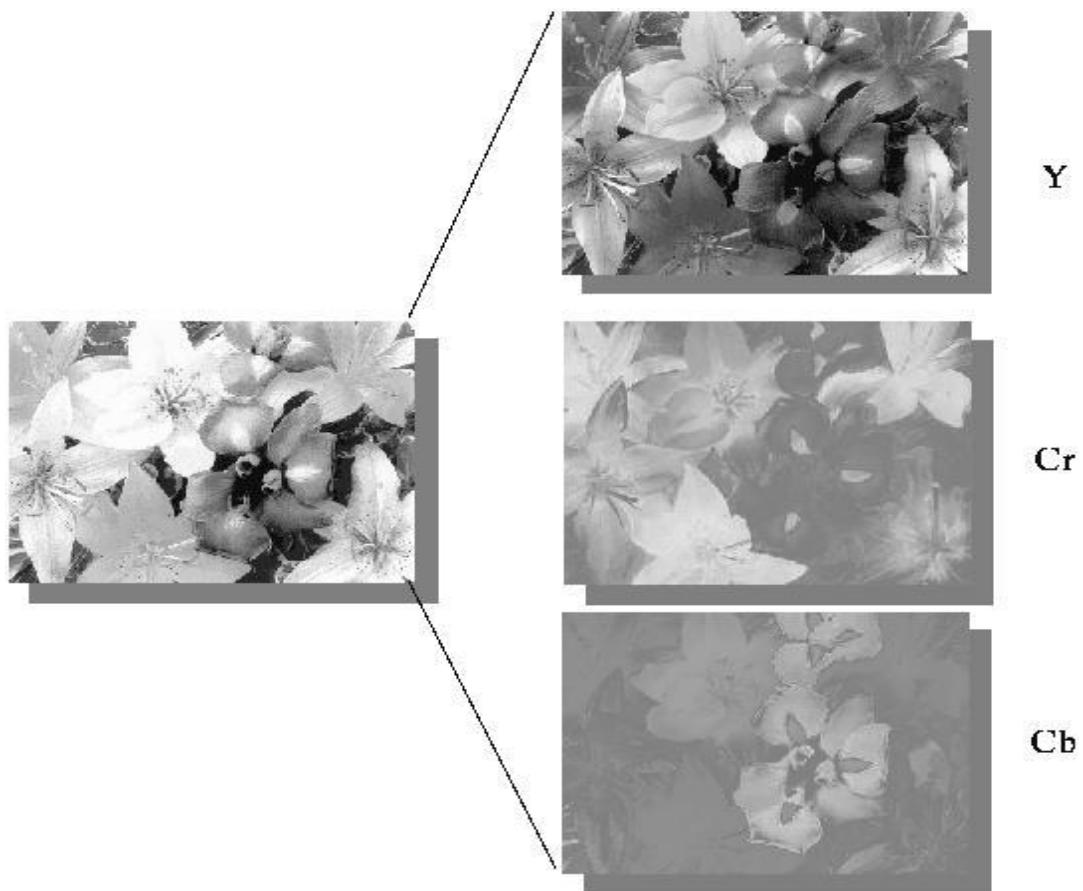


Figure 2.3 YUV components.[2]

YUV model is now used for PAL analog video; it is now also used in CCIR 601 standard for digital video.

- **YUV Transformation:**

This transformation is used to transform from RGB color model to YUV color model and vice versa .

- The YUV Transform:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2.2)$$

The YUV Inverse Transform

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.4 \\ 1 & -0.344 & -0.714 \\ 1 & 1.77 & 0 \end{bmatrix} \begin{bmatrix} Y \\ U \\ V \end{bmatrix} \quad (2.3)$$

2.3 Standards of some basic media formats:

2.3.1 Bitmap (BMP) standard Format:

Bitmaps are over any place. Bitmaps are not very popular on the Web, however. Since most bitmaps are not compressed, they tend to be larger than they would be if they were in GIF or JPEG format.

Bitmap is the simplest image format.

A bitmap file consists of four parts: the bitmap file header (see table 2.2)[3], the bitmap info header (see table 2.2)[3], the color table, and the bits that make up the image.

Table 2.1 Format of the Bitmap File Header

<i># of Bytes</i>	<i>Type</i>	<i>Description</i>
2	Character	File type (should be the characters 'B' and 'M')
4	32-bit Int	Total size of the file in bytes
2	16-bit Int	Reserved

2	16-bit Int	Reserved
4	32-bit Int	Byte-offset in file where the actual bitmap bits begin

Table 2.2 Format of the Bitmap Info Header

<i># of Bytes</i>	<i>Type</i>	<i>Description</i>
4	32-bit Int	Size (in bytes) of the info header
4	32-bit Int	Width of bitmap (in pixels)
4	32-bit Int	Height of bitmap (in pixels)
2	16-bit Int	Number of bitplanes (should be 1)
2	16-bit Int	Number of bits per pixel (should be 1, 4, 8, or 24)
4	32-bit Int	Type of compression used
4	32-bit Int	Actual number of bytes in bitmap (only necessary if compression is used)
4	32-bit Int	Number of horizontal pixels per meter (used for scaling)
4	32-bit Int	Number of vertical pixels per meter (used for scaling)
4	32-bit Int	Number of colors actually used
4	32-bit Int	Number of colors that are really important (helps when reducing the number of colors)

The file type in the bitmap file header allows a program to make sure that this is a bitmap file before proceeding. If it doesn't start with "BM," it isn't a bitmap file. The byte offset for the bitmap bits is important, because there may be some padding between the headers and the actual bits. You need to know how much padding to skip over. And after this header then it followed by the bits that make up the image in RGB style if the image is in true color style and in one byte style if the image is a gray one.

2.3.2 JPEG standard format:

2.3.2.1 Description:

JPEG is a shortcut for (Joint Photographic Experts Group)[3], JPEG format is a very popular on the Web, because it is small size, compressed and reasonable quality[4].

The Jpeg file format consists of two parts:

- 1- JPEG file header
- 2- JPEG data segment

1) JPEG Header:

- Header (2 bytes): \$ff, \$d8 (SOI) (these two identify a JPEG/JFIF file)
- For JFIF files, an APP0 segment is immediately following the SOI marker,
- Any number of "segments" (similar to IFF chunks), see below
- Trailer (2 bytes): \$ff, \$d9 (EOI)

2) Segment format:

- Header (4 bytes):
 - \$ff identifies segment
 - n type of segment (one byte)
 - sh, sl size of the segment, including these two bytes
 - High byte first, low byte last!
- Contents of the segment maximum bytes are 65533 bytes.

2.3.2.2 The JPEG compression standards:

The JPEG encodings steps are[3]:

- 1) The fine transformation in color space: [R G B] -> [Y Cb Cr] For more information see section 2.2.3 (YUV Transformation)
- 2) Sampling:

The JPEG standard takes into account the fact that the eye seems to be more sensitive at the luminance of a color than at the nuance of that color.

(The white-black view cells have more influence than the day view cells).

So, on most JPGS, luminance is taken in every pixel while the chrominance is taken as a medium value for a 2x2 block of pixels.

Note that it is not necessarily that the chrominance to be taken as a medium value for a 2x2 block , it could be taken in every pixel, but good compression results are achieved this way, with almost no loss in visual perception of the new sampled image.

3) Level shift:

All 8-bit unsigned values (Y,Cb,Cr) in the image are "level shifted": they are converted to an 8-bit signed representation, by subtracting 128 from their value.

4) The 8 x 8 Discrete Cosine Transform (DCT)

The image is break into 8x8 blocks of pixels, then for each 8x8 block is applied the DCT transform. Note that if the X dimension of the original image is not divisible by 8, the encoder should make it divisible, by completing the remaining right columns (until X becomes a multiple of 8) with the right-most column of the original image.

Similar, if the Y dimension is not divisible by 8, the encoder should complete the remaining lines with the bottom-most line of the original image.

The 8x8 blocks are processed from left to right and from top to bottom. (See section 3.2; the discrete cosine transform).

5) The zigzag reordering of the 64 DCT coefficients:

So, after we performed the DCT transform over a block of 8x8 values, we have a new 8x8 block. Then, this 8x8 block is traversed in zigzag manner (see figure 3.7).

6) Quantization:

At this stage, we have a sorted vector with 64 values corresponding to the amplitudes of the 64 spatial frequencies present in the 8x8 block.

These 64 values are quantized: Each value is divided by a dividend specified in a vector with 64 values --- the quantization table, and then it's rounded to the nearest integer.

7) The Zero Run Length Coding (RLC):

Now we have the quantized vector with a lot of consecutive zeroes. We can exploit this by run length coding the consecutive zeroes.

8) The final step apply Huffman coding

First an IMPORTANT note : Instead of storing the actual value , the JPEG standard specifies that we store the minimum size in bits in which we can keep that value (it's called the category of that value) and then a bit-coded representation.

2.3.3 AVI standard format:

2.3.3.1 Standard AVI file format:

AVI format is one of the famous file formats released by Microsoft AVI Files are a special case of RIFF files. RIFF is the Resource Interchange File Format. This is a general purpose format for exchanging multimedia data types that was defined by Microsoft and IBM .

AVI and Windows Bitmaps (DDB, DIB...):

Microsoft Windows represents bitmapped images internally and in files as Device Dependent Bitmaps (DDB), Device Independent Bitmaps (DIB), and DIB Sections. Uncompressed 'DIB' AVI files represent video frames as

DIB's. Various multimedia API's that work with AVI use Windows bitmapped images.

RIFF files are built from:

(1) RIFF Form Header:

'RIFF' (4 byte file size) 'xxxx' (data), where 'xxxx' identifies the specialization (or form) of RIFF.

'AVI' for AVI files, where the data is the rest of the file. The data is comprised of chunks and lists. Chunks and lists are defined immediately below.

(2) A Chunk:

(4 byte identifier) (4 byte chunk size) (Data)

The 4 byte identifier is a human readable sequence of four characters such as 'JUNK' or 'idx1'.

(3) A List:

'LIST' (4 byte list size) (4 byte list identifier) (Data)

Where the 4 byte identifier is a human readable sequence of four characters such as 'rec' or 'movi', where the data is comprised of LISTS or CHUNKS.

There is only one RIFF chunk per file. The chunk is a group of frames with their data, location and information.

RIFF sub-chunks may be either LIST chunks or regular sub-chunks. The LIST chunk obeys the same structure and may have regular or LIST sub-chunks; all other RIFF sub-chunks have just an ID identifier and a size. These regular sub-chunks may not have sub-chunks.

CHAPTER 3
DISCRETE COSINE
TRANSFORM

Chapter 3

Discrete Cosine Transform

3.1 Introduction:

Transform coding constitutes an integral component of applications. contemporary image/video processing Transform coding relies on the premise that pixels in an level of correlation with their image exhibit a certain neighboring pixels[5]. Similarly in a video transmission ,system

adjacent pixels in consecutive frames show very high correlations can exploited to correlation. Consequently, thes predict the value of a pixel from its respective neighbors. A transformation is, therefore, defined to map this spatial (correlated) data into transformed uncorrelated) coefficients. Clearly, the transformation should utilize the fact that the content of an individual pixel is relatively small information pixel can be i.e., to a large extent visual contribution of a .predicted using its neighbors

Interest in multimedia and digital image processing methods emerges from two principles application areas: improvement of pictorial information for human interpretation; and processing of image data for storage, transmission, compression, and representation for autonomous machine perception . This chapter aims to define the scope of the fields that we call multimedia and image processing, discuss briefly the principle approaches used in this field and define the major color systems used in media and how to transform a media from one to another[6].

3.2 The Discrete Cosine Transform

Like other transforms, the Discrete Cosine Transform (DCT) attempts to decorrelate the image data. After decorrelation each transform coefficient can be encoded independently without losing compression efficiency. This section describes the DC

3.2.1 DCT:

The Discrete Cosine Transform, or DCT, transforms data into a format that can be easily compressed. The characteristics of the DCT make it ideally suited for image compression algorithms. These algorithms let you minimize the amount of data needed to recreate a digitized image.

DCT is a transformation that transforms a sequence of discrete signals into another data sequence $\{x_0, x_1 \dots x_{n-1}\} \rightarrow \{z_0, z_1 \dots z_{n-1}\}$. The purpose of the transform is to separate the input

sequence into relevant and irrelevant information. The irrelevant information is then suppressed by a quantizer [9]. Figure 3.1 shows how the input signal $X \{x_0, x_1 \dots x_{n-1}\}$ is transformed into the sequence $Z = \{z_0, z_1 \dots z_{n-1}\}$. The elements z_k of this sequence are then separately quantized by the quantizer q_k . This yields the data $Q_k = \{q_0, q_1 \dots q_{n-1}\}$, which is transmitted over a communication channel to the receiver.

At the receiver side the data is then dequantized to Z' where $Z' \sim Z$. The data Z' is then transformed back to X' .

Quantization will be further discussed in details in section 3.4.1.

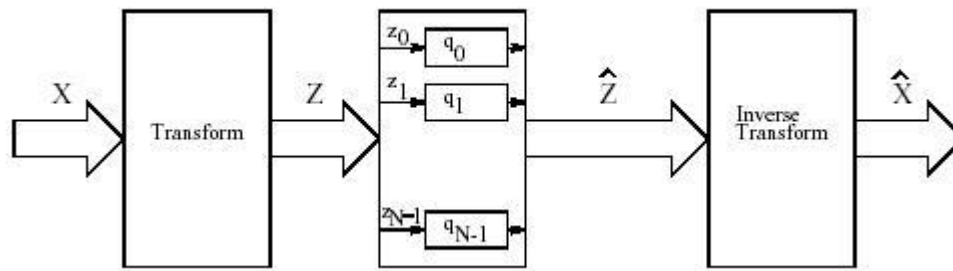


Figure 3.1: A scheme of transformation coding.[7]

The purpose of these transformation is to compact as much relevant information in few as possible signal elements. There are many kinds of transforms. One transform is the Discrete Cosine Transform (DCT), which was first introduced by Ahmed ET. Al. in 1974 [10].

The purpose of the DCT transform is that instead of processing the original samples, you work with the spatial frequencies

3.2.2 The One-Dimensional Discrete Cosine Transform

The discrete cosine transform of a list of n real numbers $s(x)$, $x = 0, \dots, n-1$, is the list of length n given by:

Mathematica Journal, 4(1), 1994, p. 81-88

$$S(u) = \frac{1}{2} \sum_{x=0}^{n-1} C(u) s(x) \cos\left(\frac{(2x+1)u\pi}{2n}\right)$$

$$x=0$$

$$n-1 \text{ } \varepsilon \text{ } u = 0$$

$$\text{where } C(u) = \begin{cases} \frac{1}{2} & \text{for } u = 0 \\ 1 & \text{otherwise} \end{cases}$$

$$= 1 \text{ otherwise}$$

3.2.3 The Two-Dimensional DCT[8]

The objective of this document is to study the efficacy of DCT on images. This necessitates the extension of ideas presented in the last section to a two-dimensional space. The 2-D DCT is a direct extension of the 1-D case .

3.3 Properties of DCT

Discussions in the preceding sections have developed a mathematical foundation for DCT.

However, the intuitive insight into its image processing application has not been presented. This section outlines (with examples) some properties of the DCT which are of particular value to image processing applications.

2.3.1. Decorrelation

2.3.2. Energy Compaction

2.3.3. Separability

2.3.4. Symmetry

2.3.5. Orthogonality

3.4 Forward Discrete Cosine Transfer (DCT)

3.4.1Features

Supports FLEX® 10KE, ACEX® 1K and APEX™ 20KE devices

- High clock speed
- Low gate count
- 8 x 8 DCT block size
- Continuous one symbol per clock cycle

- 8 bpp inputs, 11 bit output coefficients, 10 bit cosine coefficients and 15 bit internal computations precision
- No internal RAM requirements
- Internal zero-level shifting on input samples
- Low latency (82 cycles)
- Fully synchronous, without internal tri-states, with asynchronous reset
- Available in VHDL
- Inverse DCT

3.4.2Description

The discrete cosine transform (DCT) core implements the two dimensional (2D) forward DCT (2D-DCT) on an 8 x 8 block of samples. Hence, it is appropriate for DCT-based image or video encoders and can be used as a core for the JPEG, MPEG1, MPEG2, MPEG4, H.261, and H.263 standards. The core is based on the row-column computational architecture.

The DCT core is designed for reuse in ASIC and programmable logic devices. The design is fully synchronous with positive edge clocking and no internal tri-state buffers. It offers high performance, with a low gate count, and can be used in any multimedia, digital video, or digital printing application.

3.5 Forward DCT:

$$F(u) = \frac{2}{N} c(u) \sum_{x=0}^{N-1} f(x) \cos \frac{(2x+1)u\pi}{2N} \quad , u=0,1,2,\dots,N-1$$

(3.1)

IDCT:

$$F(x) = \sum_{u=0}^{N-1} c(u) f(u) \cos \frac{(2x+1)u\pi}{2N} \quad , u = 0,1,2,\dots, N-1$$

(3.2)

$$c(u) = 1/\sqrt{2}, \text{ for } u = 0$$

$$c(u) = 1, \text{ for } u = 1, 2, \dots, N-1$$

The first equation is used to transform the image from spatial domain to frequency domain, while the other one is used to transform the image from frequency domain to spatial domain.

In the last equations, N is the number of discrete signals in the input sequence.

After we have performed the DCT transform over a block of 8×8 values, we have a new 8×8 block. Then, this 8×8 block is traversed in zigzag (like figure 3.2) which indicates the order in which we traverse the bidimensional 8×8 matrix in zigzag manner.

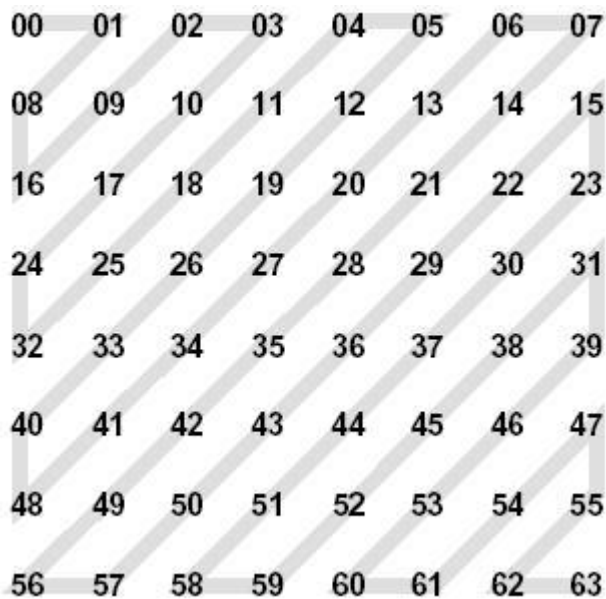


Figure 3.2 Zigzag points order for 8 x 8 matrix

To implement the transform in image, we must formulate a 2D equation.

- The formation of the equation in a 8×8 sample blocks:

The following equations are the idealized mathematical definitions of the 8x8 Forward DCT and 8x8 IDCT:

Forward DCT:

$$F(u, v) = \frac{1}{4} C(u) C(v) \left[\sum_{x=0}^7 \sum_{y=0}^7 f(x, y) * \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16} \right]$$

(3.3)

$$f(x, y) = \frac{1}{4} \left[\sum_{u=0}^7 \sum_{v=0}^7 C(u) C(v) F(u, v) * \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16} \right]$$

(3.4)

IDCT:

$$c(u), c(v) = 1/\sqrt{2}, \text{ for } u = v = 0$$

$$c(u), c(v) = 1, \text{ for } u, v = 1, 2, \dots, N-1$$

3.6 Fast Discrete Cosine algorithm:

Hou's Fast Discrete Cosine Transform Algorithm [10] (FDCT) is numerically stable, fast and recursive. Similar to the Cooley-Tukey FFT algorithm, it generates the next larger DCT from two identical, smaller DCTs. This deviates from direct factorization algorithms that factor the desired N-pt DCT matrix. In that, the higher order matrices are generated from lower order DCT matrices instead. Refer to Hou's paper for a tutorial on the DCT in general and his algorithmic implementation.

CHAPTER 4
AUTHENTICATION AND
DIGITAL SIGNATURE

Chapter 4

Authentication And Digital Signature

4.1 Introduction

For almost 150 years, the U.S. Government Printing Office (GPO) has been the official disseminator of Government documents and has assured users of their authenticity.

With introduction of the computer, the need for automated tools for protecting files and other information stored on the computer became evident. This is especially the case for a shared system, such as a time-sharing system, and the need is even more acute for systems that can be accessed over a public telephone network, data network, or the internet.

In the 21st century, the increasing use of electronic documents poses special challenges in verifying authenticity, because digital technology makes such documents easy to alter or copy, leading to multiple non-identical versions that can be used in unauthorized or illegitimate ways.

To help meet the challenge of the digital age, GPO has begun implementing digital signatures to certain electronic documents on *GPO Access* that not only establish GPO as the trusted information disseminator, but also provide the assurance that an electronic document has not been altered since GPO disseminated it.

The visible digital signatures on online PDF documents serve the same purpose as handwritten signatures or traditional wax seals on printed documents. A digital signature, viewed

through the GPO Seal of Authenticity, verifies document integrity and authenticity on GPO online Federal documents, at no cost to the customer.

Digital watermarking has been proposed as a solution to the problem of copyright protection of multimedia data in a networked environment. It makes possible to tightly associate to a digital document a code allowing the identification of the data creator, owner, authorized consumer, and so on.

4.2 Authentication

The authentication is concerned with assuring that a communication is authentic. In the case of a single message, such as a warning or alarm signal, the function of the authentication service is to assure the recipient that the message is from the source that it claims to be from. In the case of an ongoing interaction, such as the connection of a terminal to a host, two aspects are involved. First, at the time of connection initiation, the service assures that the two entities are authentic, that is, that each is the entity that it claims to be. Second, the service must assure that the connection is not interfered with in such a way that a third party can masquerade as one of the two legitimate parties for the purposes of unauthorized transmission or reception.

The process of identifying an individual usually based on a username and password. In security systems, authentication is distinct from authorization, which is the process of giving individuals access to system objects based on their identity.

Authentication merely ensures that the individual is who he or she claims to be, but says nothing about the access rights of the individual.[11]

4.2.1: Authentication factors and identity

The ways in which someone may be authenticated fall into three categories, based on what are known as the factors of authentication: something you know, something you have, or something you are. Each **authentication factor** covers a range of elements used to authenticate or verify a person's identity prior to being granted access, approving a transaction request, signing a document or other work product, granting authority to others, and establishing a chain of authority. Security research has determined that for a positive identification, elements from at least two, and preferably all three, factors be verified. The three factors (classes) and some of elements of each factor are:

- the **ownership factors**: Something the user **has** (e.g., wrist band, ID card, security token, software token, phone, or cell phone)
- the **knowledge factors**: Something the user **knows** (e.g., a password, pass phrase, or personal identification number (PIN), challenge response (the user must answer a question))
- the **inherence factors**: Something the user **is** or **does** (e.g., fingerprint, retinal pattern, DNA sequence (there are assorted definitions of what is sufficient), signature, face, voice, unique bio-electric signals, or other biometric identifier).

These factors can be used alone, or they can be combined to build a stronger authentication strategy in what is known as two-factor or multi authentication.

4.2.1.1: Two-factor authentication

When elements representing two factors are required for identification, the term *two-factor authentication* is applied. . e.g. a bankcard (something the user **has**) and a PIN (something the user **knows**). Business networks may require users to provide a password (knowledge factor) and a random number from a security token (ownership factor). Access to a very high security system might require a mantrap screening of height, weight, facial, and fingerprint checks (several inherence factor elements) plus a PIN and a day code (knowledge factor elements), but this is still a two-factor authentication.

Authentication can be conducted through the use of logon passwords, single sign-on (SSO) systems, biometrics, digital certificates and a public key infrastructure (PKI).

4.2.2: Levels of Proof:

There are four levels of proof that people are indeed who they say they are. None of them are entirely foolproof, but in order of least to most secure, they are:

1 - What You Know

Passwords are widely used to identify a user, but only verify that somebody knows the password.

2 - What You Have

Digital certificates in the user's computer add more security than a password, and smart cards verify that users have a

physical token in their possession, but both laptops and smart cards can be stolen.

3 - What You Are

Biometrics such as fingerprints and iris recognition are more difficult to forge, but you have seen such systems fooled in the movies all the time.

4 - What You Do

Dynamic biometrics such as hand writing a signature and voice recognition are the most secure; however, replay attacks can fool the system.

4.3 Digital signature

A **digital signature** or **digital signature scheme** is a mathematical scheme for demonstrating the authenticity of a digital message or document. A valid digital signature gives a recipient reason to believe that the message was created by a known sender, and that it was not altered in transit.

Digital signatures are commonly used for software distribution, financial transactions, and in other cases where it is important to detect forgery and tampering.

Digital signatures are often used to implement electronic signatures, a broader term that refers to any electronic data that carries the intent of a signature, but not all electronic signatures use digital signatures. In some countries, including the United States, India, and members of the European Union, electronic signatures have legal significance. However, laws concerning electronic signatures do not always make clear whether they are digital

cryptographic signatures in the sense used here, leaving the legal definition, and so their importance, somewhat confused.

Digital signatures employ a type of asymmetric cryptography. For messages sent through an insecure channel, a properly implemented digital signature gives the receiver reason to believe the message was sent by the claimed sender.[12]

How It Works:

Assume you were going to send the draft of a contract to your lawyer in another town. You want to give your lawyer the assurance that it was unchanged from what you sent and that it is really from you.

1. You copy-and-paste the contract (it's a short one!) into an e-mail note.
2. Using special software, you obtain a message hash (mathematical summary) of the contract.
3. You then use a private key that you have previously obtained from a public-private key authority to encrypt the hash.
4. The encrypted hash becomes your digital signature of the message. (Note that it will be different each time you send a message.)

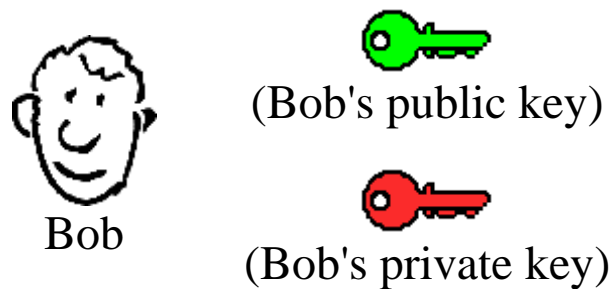
At the other end, your lawyer receives the message.

1. To make sure it's intact and from you, your lawyer makes a hash of the received message.
2. Your lawyer then uses your public key to decrypt the message hash or summary.

If the hashes match, the received message is valid.

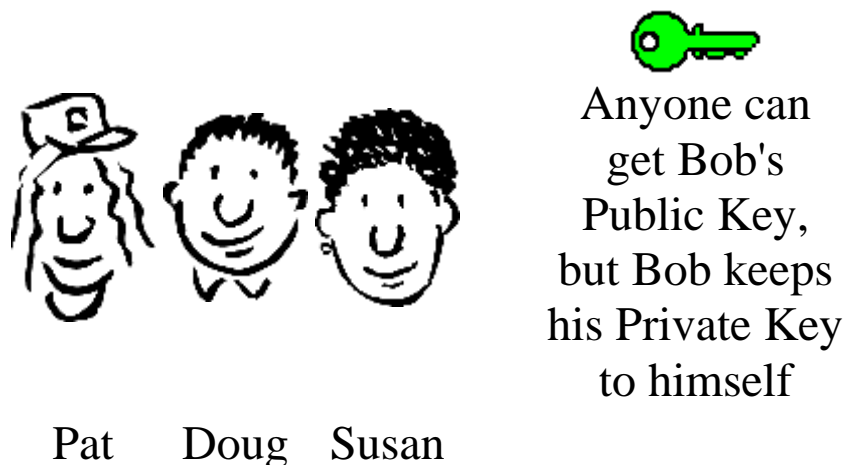
What is a Digital Signature?

An introduction to Digital Signatures, by David Youd



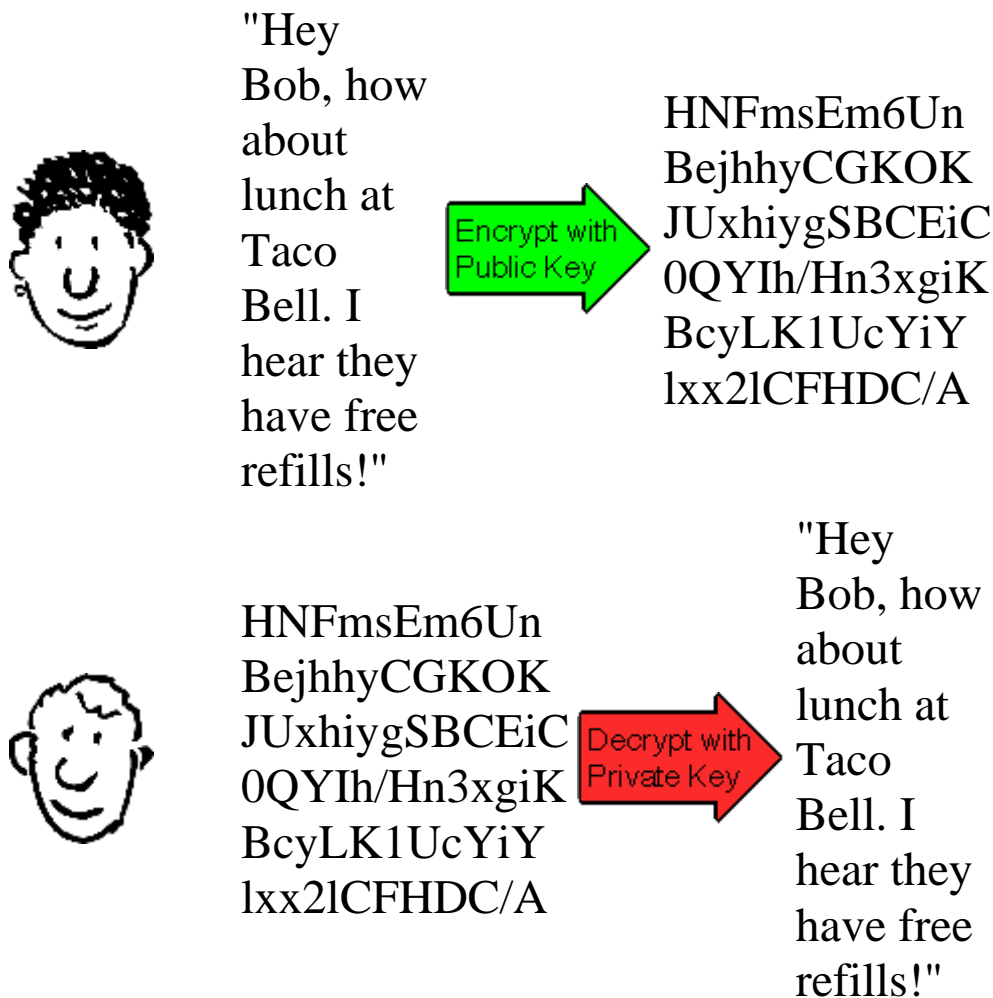
Bob has been given two keys. One of Bob's keys is called a Public Key, the other is called a Private Key.

Bob's Co-workers:



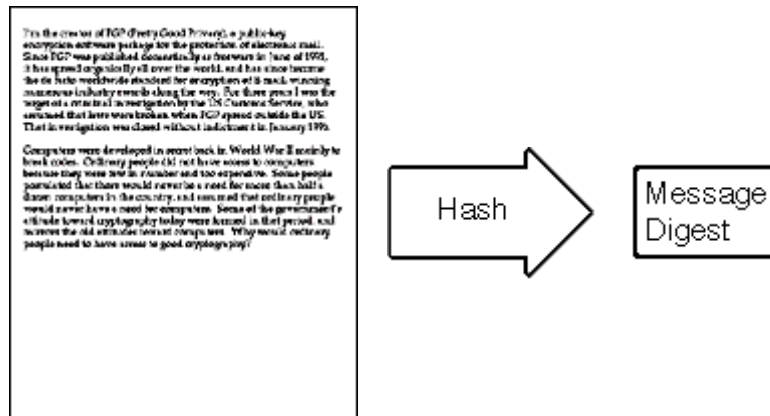
Bob's Public key is available to anyone who needs it, but he keeps his Private Key to himself. Keys are used to encrypt information. Encrypting information means "scrambling it up", so that only a person with the appropriate key can make it readable again. Either one of Bob's two keys can encrypt data, or the other key can decrypt that data.

Susan (shown below) can encrypt a message using Bob's Public Key. Bob uses his Private Key to decrypt the message. Any of Bob's coworkers might have access to the message Susan encrypted, but without Bob's Private Key, the data is worthless.



With his private key and the right software, Bob can put digital signatures on documents and other data. A digital

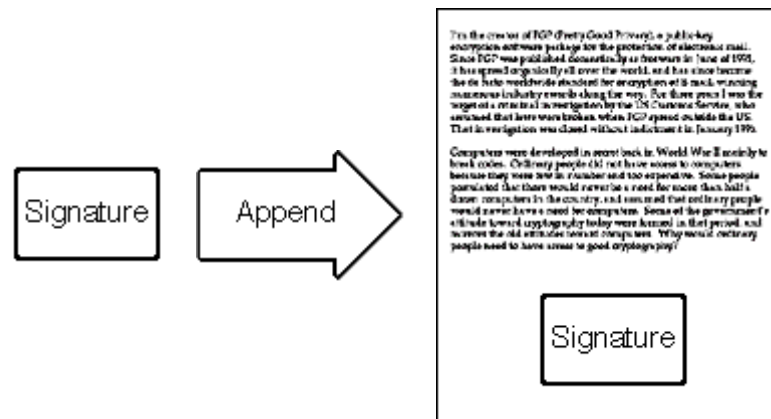
signature is a "stamp" Bob places on the data which is unique to Bob, and is very difficult to forge. In addition, the signature assures that any changes made to the data that has been signed can not go undetected.



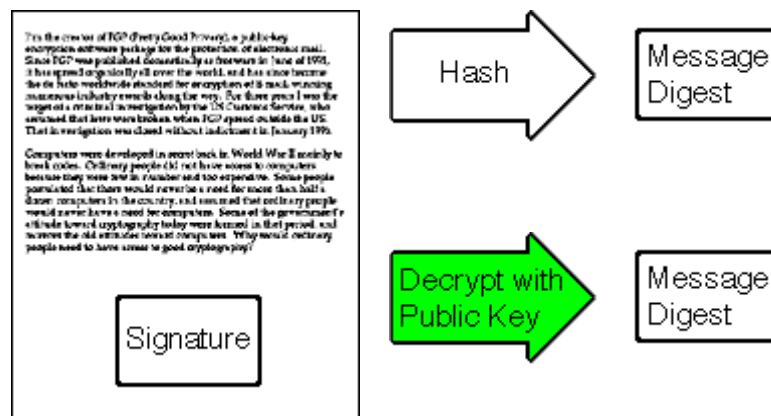
To sign a document, Bob's software will crunch down the data into just a few lines by a process called "hashing". These few lines are called a message digest. (It is not possible to change a message digest back into the original data from which it was created.)



Bob's software then encrypts the message digest with his private key. The result is the digital signature.



Finally, Bob's software appends the digital signature to document. All of the data that was hashed has been signed.



Bob now passes the document on to Pat. [12]



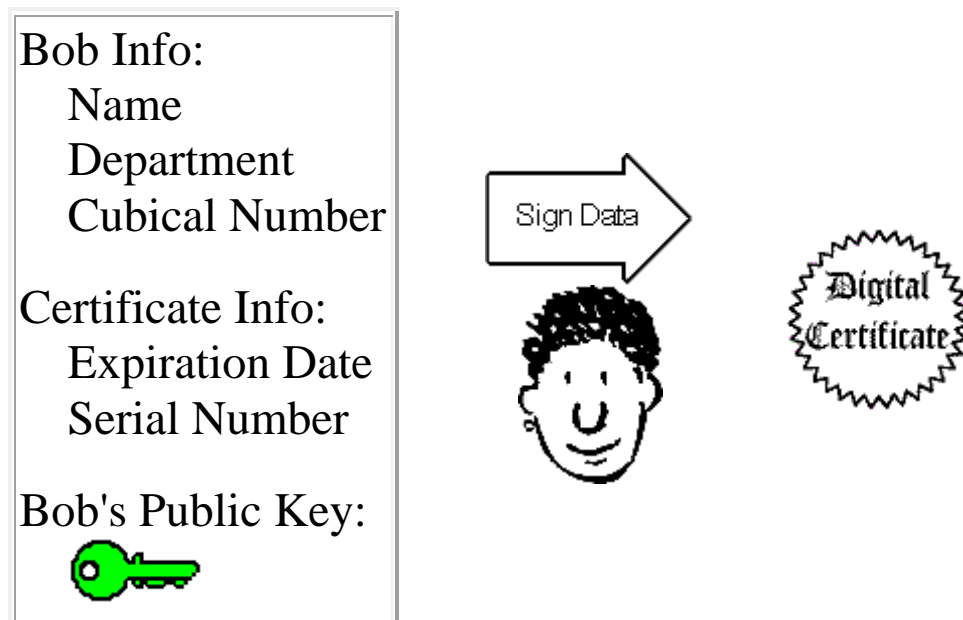
First, Pat's software decrypts the signature (using Bob's public key) changing it back into a message digest. If this worked, then it proves that Bob signed the document, because only Bob has his private key. Pat's software then hashes the document data into a message digest. If the message digest is the same as the message digest created when the signature was decrypted, then Pat knows that the signed data has not been changed.

Plot complication...



Doug (our disgruntled employee) wishes to deceive Pat. Doug makes sure that Pat receives a signed message and a public key that appears to belong to Bob. Unbeknownst to Pat, Doug deceitfully sent a key pair he created using Bob's name. Short of receiving Bob's public key from him in person, how can Pat be sure that Bob's public key is authentic?

It just so happens that Susan works at the company's certificate authority center. Susan can create a digital certificate for Bob simply by signing Bob's public key as well as some information about Bob.

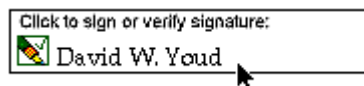


Now Bob's co-workers can check Bob's trusted certificate to make sure that his public key truly belongs to him. In fact, no one at Bob's company accepts a signature for which there does not exist a certificate generated by Susan. This gives Susan the power to revoke signatures if private keys are compromised, or no longer needed. There are even more widely accepted certificate authorities that certify Susan.

Let's say that Bob sends a signed document to Pat. To verify the signature on the document, Pat's software first uses

Susan's (the certificate authority's) public key to check the signature on Bob's certificate. Successful de-encryption of the certificate proves that Susan created it. After the certificate is de-encrypted, Pat's software can check if Bob is in good standing with the certificate authority and that all of the certificate information concerning Bob's identity has not been altered.

Pat's software then takes Bob's public key from the certificate and uses it to check Bob's signature. If Bob's public key de-encrypts the signature successfully, then Pat is assured that the signature was created using Bob's private key, for Susan has certified the matching public key. And of course, if the signature is valid, then we know that Doug didn't try to change the signed content.



Although these steps may sound complicated, they are all handled behind the scenes by Pat's user-friendly software. To verify a signature, Pat need only click on it.

So, Digital signatures are equivalent to traditional handwritten signatures in many respects; properly implemented digital signatures are more difficult to forge than the handwritten type.

Digital signature schemes in the sense used here are cryptographically based, and must be implemented properly to be effective. Digital signatures can also provide non-repudiation, meaning that the signer cannot successfully claim they did not sign a message, while also claiming their private key remains secret; further, some non-repudiation schemes offer a time stamp for the digital signature, so that even if the private key is exposed, the signature is valid nonetheless. Digitally signed messages may be anything representable as a bitstring: examples include electronic

mail, contracts, or a message sent via some other cryptographic protocol.

Digital signatures are easily transportable, cannot be imitated by someone else, and can be automatically time-stamped. The ability to ensure that the original signed message arrived means that the sender cannot easily repudiate it later.

A digital signature can be used with any kind of message, whether it is encrypted or not, simply so that the receiver can be sure of the sender's identity and that the message arrived intact. A digital certificate contains the digital signature of the certificate-issuing authority so that anyone can verify that the certificate is real.

This project proposes an authentication scheme for BMP images based on digital signature and watermarking. It can detect and locate malicious manipulations made to the image, and verify the ownership of the image at the same time. The algorithm uses the invariance of the order relationship between two DCT coefficients before and after BMP compression to embed image content dependent watermark, therefore the watermark can survive the BMP lossy compression.

Since the scheme is based on the use of signatures like the use of **User Name, Ip Address, Mac Address, Date Time**, it is believed to be secure to the extent that cryptography is believed to be and it make the sender and receiver make sure that the message secure from any attack . Theoretical analysis and experimental results show that the proposed scheme has the desired property and good performance for image authentication.

CHAPTER 5

DESIGN AND

IMPLEMENTATION

Chapter 5

Design and implementation

5.1 Introduction:

However, the use of traditional ciphering techniques is not preferable as this philosophy will raise cryptanalyst doubts about the content of this information, with the increased technology of digital computer system and the deployment of cryptanalysis researches, it will take the crypt analyst few minutes or even hours to extract the plain text from your ciphered text.

Even if the ciphered message was not decrypted, it's possible for the cryptanalyst to damage the message by making an active attack on this message. Hence, Cryptography only is not enough.

The aim of this proposed system is to introduce protection for the information through embedding it into media container; this is known as ***Data Hiding***. Data hiding seems to be the only potential encryption technology to provide protection even data is decrypted.

This chapter will explain in details the design and implementation of the system. This includes the design of the over all system, the design of the main components of the system supported by block diagrams. The implementation of the system will also be introduced and illustrated by means of flow charts and pseudo code. We

will support this by introducing snapshots taken from the real system. The system integration will show how the system components will be integrated together to construct the over all system.

At the end of this chapter, we will state the features and requirements of this system. Suggestions for future improvements will be provided.

5.2 System Design:

The secure data hiding system is a fully functional Information Security System, and thus provides the following:

1. Data hiding system that is used to hide the plaintext into a media (either image or video) in the frequency domain. The same system is used to extract the hidden data back.
2. Authentication system that uses cryptography for authenticating users. System user will authenticate himself to the other side by encrypting a sign at the header of the sent file with his private key. This can be viewed as a digital signature system.

Figure 5.1 shows the context diagram of the secure data hiding system and the interaction between its major components.

The context diagram introduces a high level overview of the system and the basic relationships between its components.

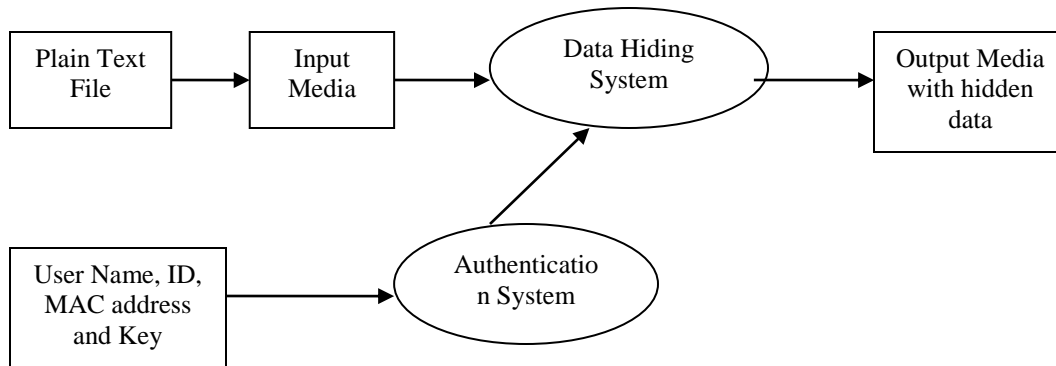


Figure 5.1 Context Diagram for Secure Data Hiding System[19].

. Data hiding hides data coming from both authentication systems into media container. The output of this system is a media file with hidden data.

The following sub sections discuss the design of the components of the system.

5.2.1 Data Hiding System Design:

The data hiding system is used to hide binary data in a media container, either image or video. Figure 5.2 shows a block diagram for data hiding subsystem.

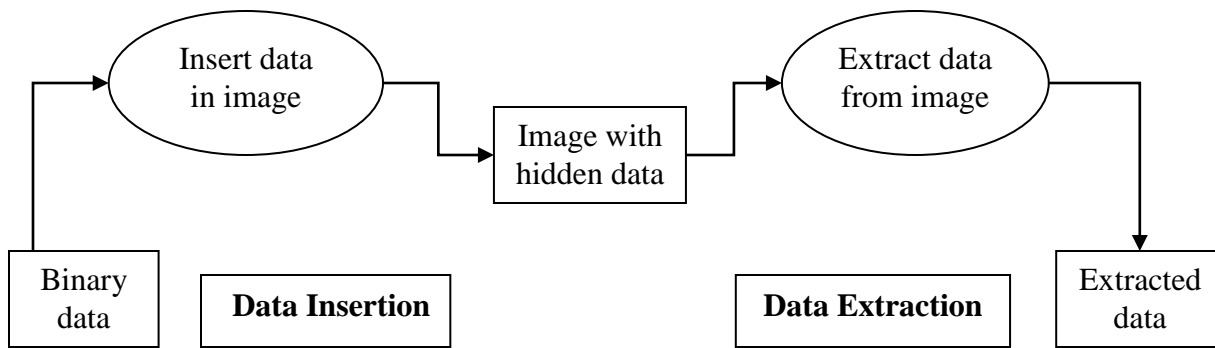


Figure 5.2 Block diagram for data hiding subsystem[19].

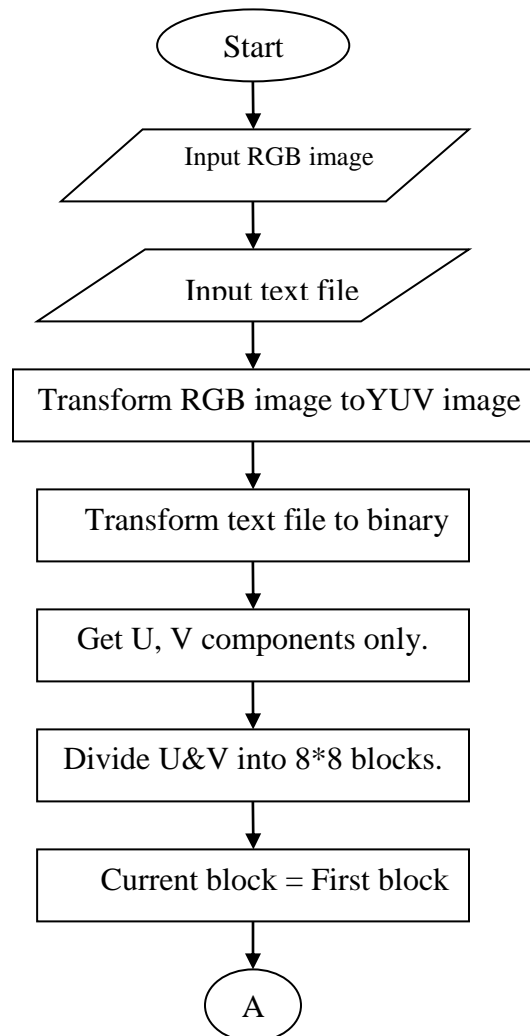
As depicted in the figure, the data is inserted in either the image. At the other side, the receiver extracts the data from the image.

5.3 System Implementation:

5.3.1 Data Hiding System Implementation:

Data hiding is performed either in image or video. Figure 5.3 shows the flow chart of hiding data in an image; to insert stream of binary data in an image, the image is first transformed from RGB color mode to YUV color mode. The equation used to perform this transformation is introduced in section 2.2. The data is hidden only in the U and V components in frequency domain. So, the second step is to divide the U and V components into 8*8 blocks to be transformed from spatial

domain to frequency domain. This is done by using the DCT or FDCT transformation. We hide the data in the U and V matrices in special positions generated by a random bit generator. After hiding process, the block is transformed back to the spatial domain using IDCT or IFDCT transform. The process continues until end of blocks or insertion of all data. Finally, the image is transformed back to the RGB color mode.



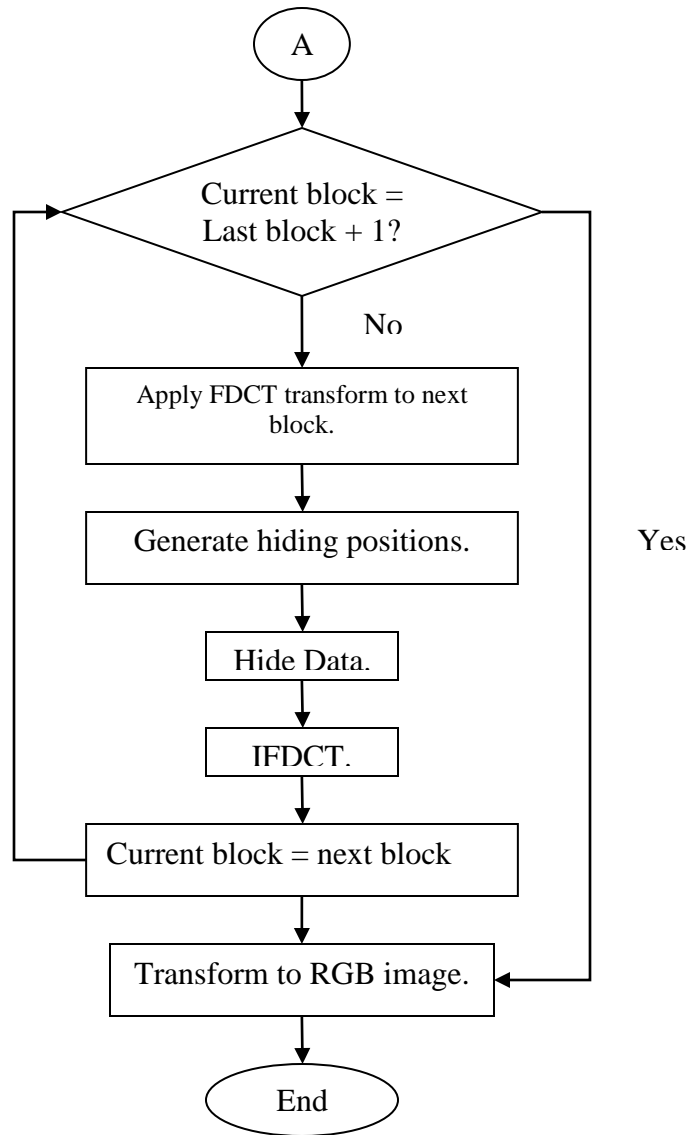
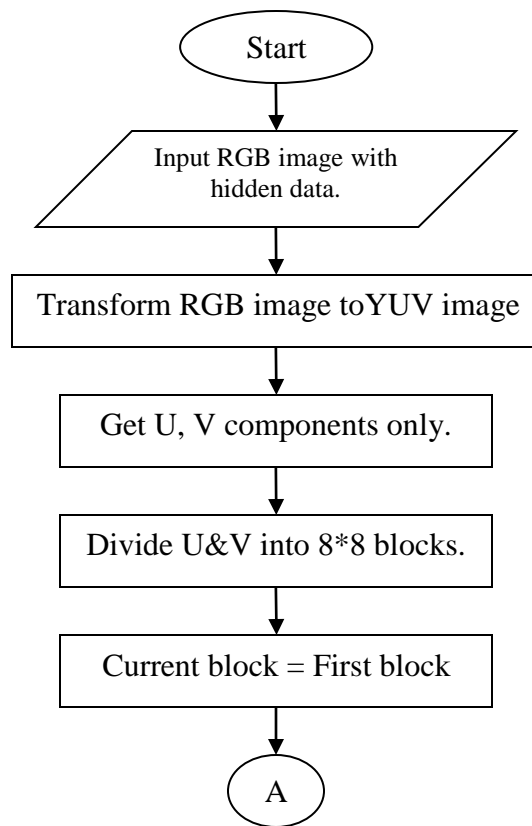


Figure 5.3 flow chart of hiding text data in a text file into an RGB image[19].

5.3.2 Data Extracting System Implementation:

The extraction process has a similar flow cart as the hiding process. The only change is that the Hide Data block is replaced by Extract Data block.



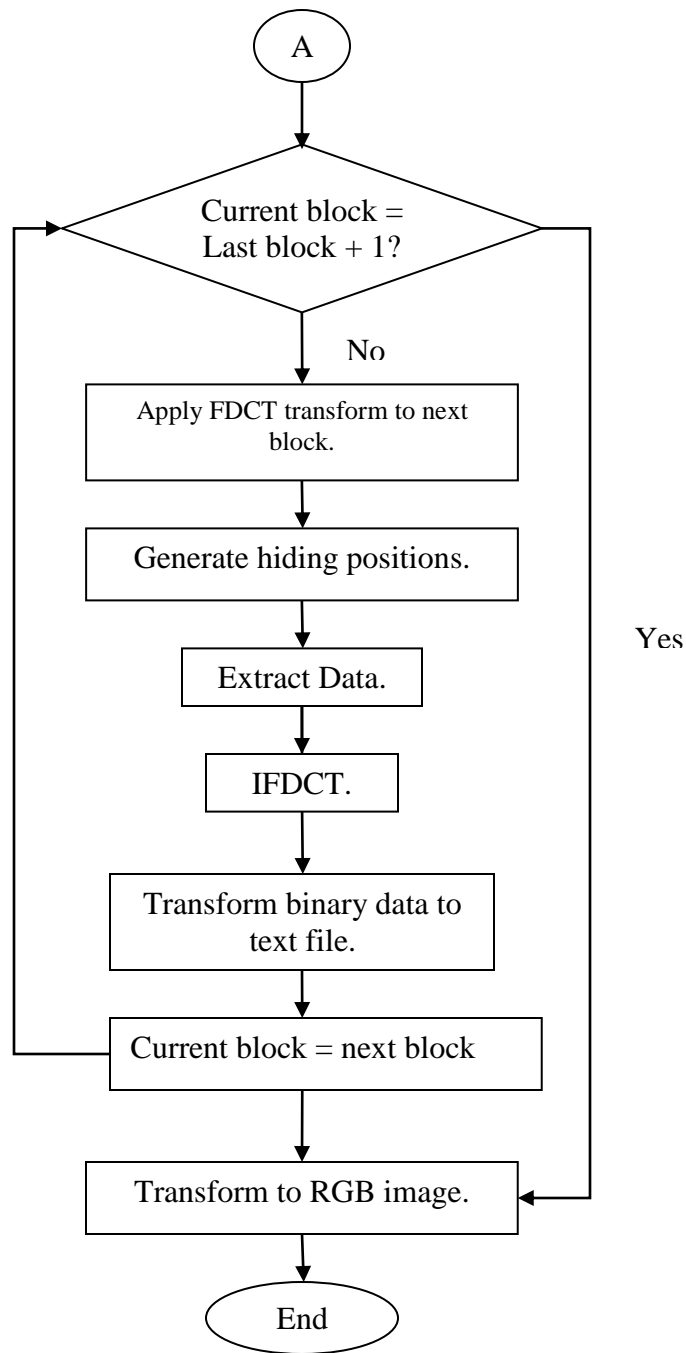


Figure 5.4 flow chart of extracting text data in a text file from RGB image[19].

The RGB to YUV transformation and its inverse is done by the following pseudo-codes:

Begin Transform-to-YUV(image-vector):[13]

$$Y = 0.299 * R + 0.587 * G + 0.114 * B.$$

$$U = -0.169 * R - 0.331 * G + 0.500 * B.$$

$$V = 0.500 * R - 0.419 * G - 0.081 * B.$$

End Transform-to-YUV(image-vector).

Begin Transform-to-RGB(image-vector):

$$R = Y + 1.4 * V.$$

$$G = Y - 0.344 * U - 0.714 * V.$$

$$B = Y + 1.77 * U.$$

End Transform-to-RGB(image-vector).

The transformation from spatial domain to frequency domain is performed by the fast discrete cosine transform (FDCT) its inverse (IFDCT).

Generation of hiding positions is done using the key generator. The following pseudo-code defines the hiding algorithm of binary data into DCT blocks:

Begin InsertIntoBlock(Block):

```
Positions=GetRandomPositions ();
norep=0;
While K < NoBitsPerBlock
    if BitPosition=BufferLength*8 then
        if End of Data then return;
        Fill Buffer With Data;
        BitValue=GetBit(BitPosition);
        If BitValue=1 then
```

```
    Block[position[k]]=(Quantizer[position[k]]/2)+Alpha;
    else
        Block[position[k]]=0;
    norep++;
    if norep=NoRepeat then
        BitPos++;
        norep=0;
```

End InsertIntoBlock(Block).

The extraction process has a similar flow cart as the insertion. The only change is that the Hide Data block is replaced by Extract Data block. The pseudo-code for data extraction is as follows:

Begin ExtractFromBlock(Block)

```
BitValue=0; QuanValue=0; norep=0;
k=0;
Loop While K < NoBitsPerBlock
    if BitBos=BufferLength*8 then
        Write Buffer To Output File;
        if End of Data then return;
    BitValue+=Block[position[k]];
    QuanValue+=Quantizer[position[k]];
    norep++;
    if norep=NoRepeat then
        if BitValue/norep>= (QuanValue/(norep*4))
            then SetBit(BitPos,true);

        else
            SetBit(BitPos,false);
        BitValue=0;
        QuanValue=0; norep=0; BitPos++;
```

End ExtractFromBlock(Block)

5.4 User Interface:

About us:



Figure 5.5 shows the team members.

For hiding data:

1: window

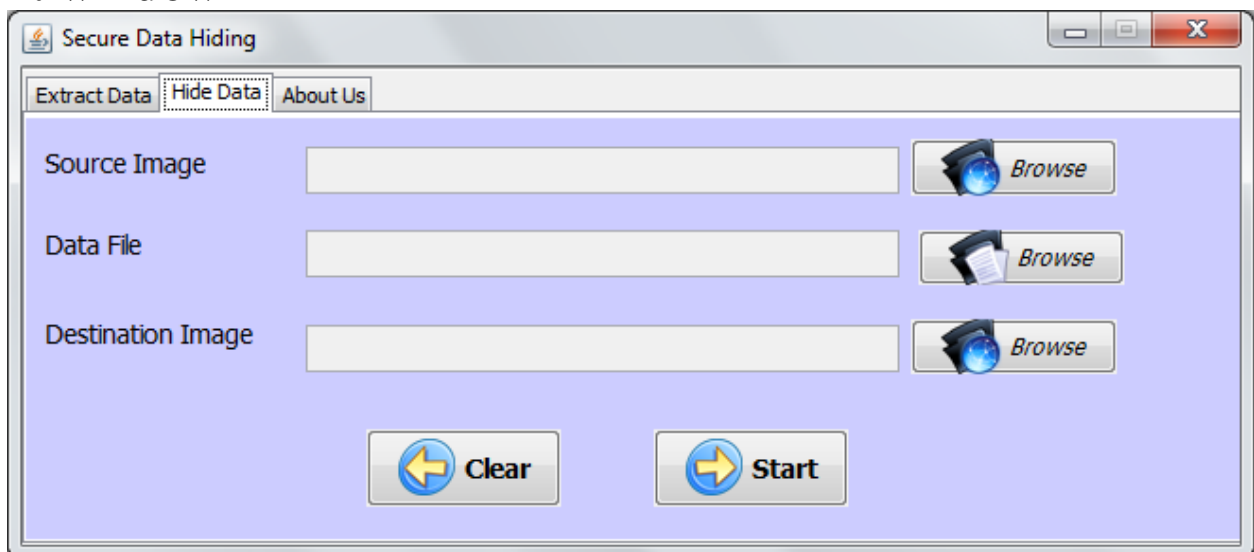


Figure 5.6 shows the window for hiding data in the image.

2: selecting image and text

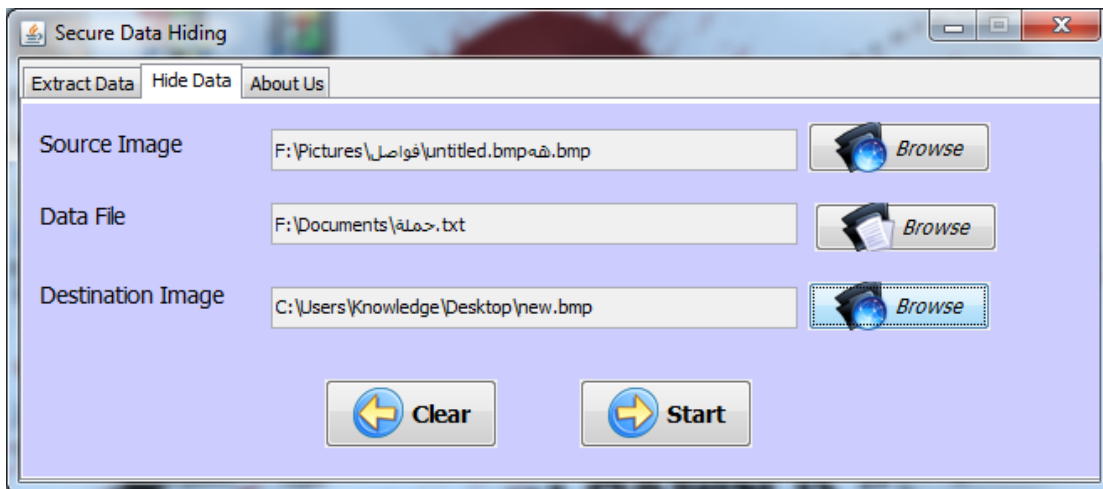


Figure 5.7 shows the window for selecting original image and text file needed to be hidden and the place of the resulted image.

3: while processing

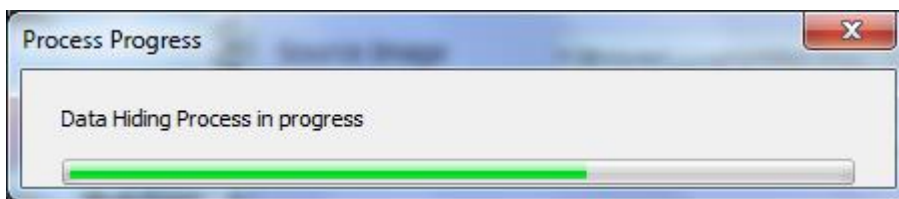


Figure 5.8 shows that the hiding is in progress.

4: After complete processing

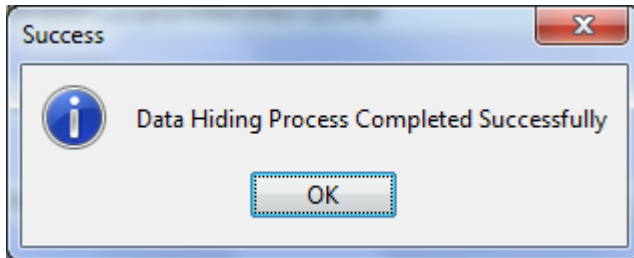


Figure 5.9 shows that the hiding completed successfully

5: image comparison



A: The Original Image

B : Image After Hiding

Figure 5.10 shows a comparison between image before and after hiding data.

For Extracting Data:

1:window

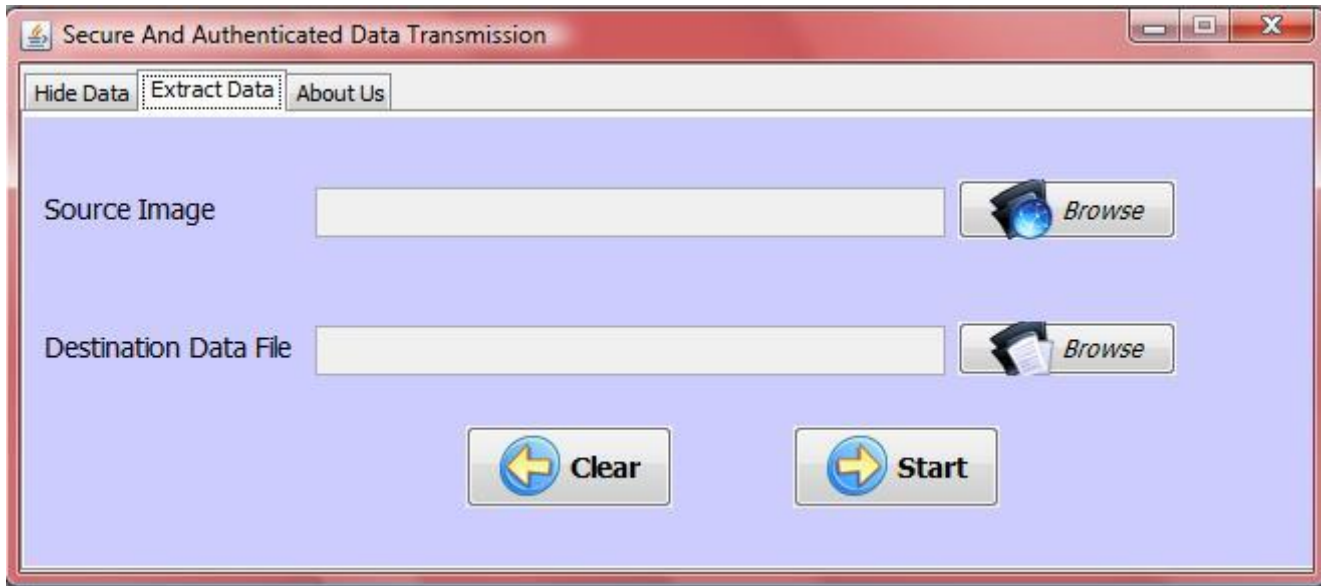


Figure 5.11 shows the window for Extracting data from the image .

2: selecting image with hidden data

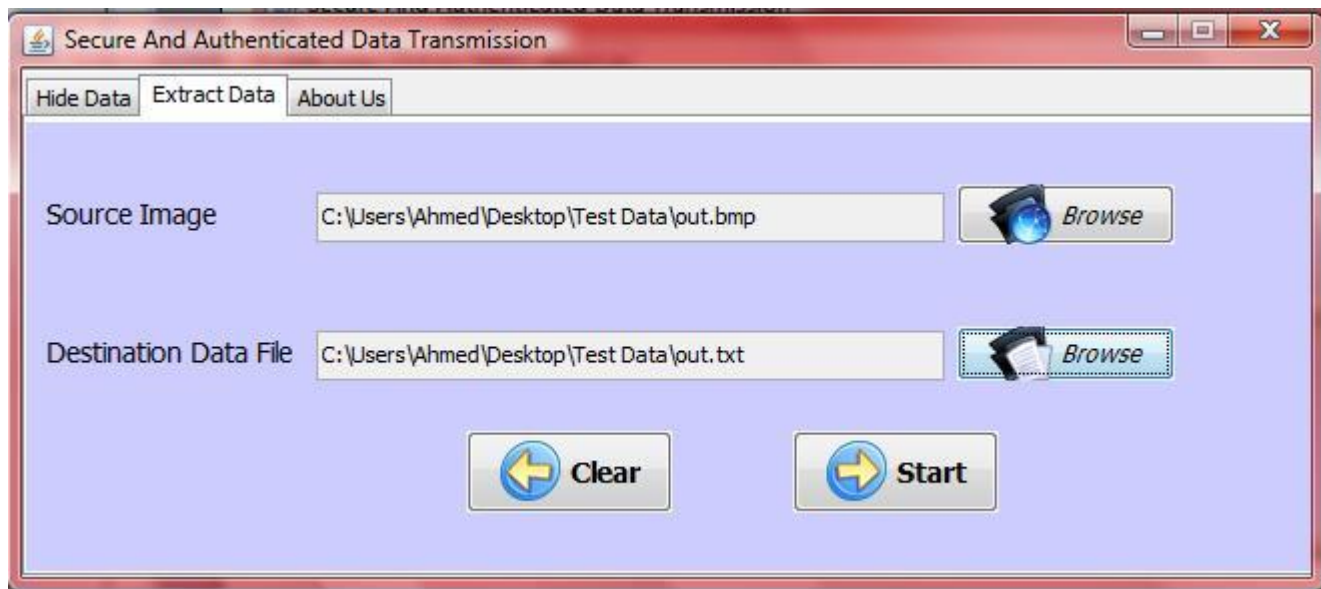


Figure 5.12 shows the window for selecting image with hidden data and the name of the new text file.

3:After extraction

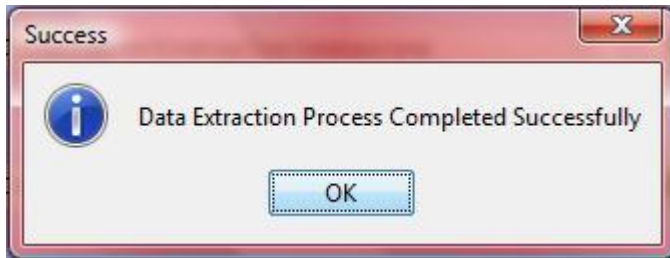


Figure 5.13 shows that the extracting completed successfully

4: Authentication message

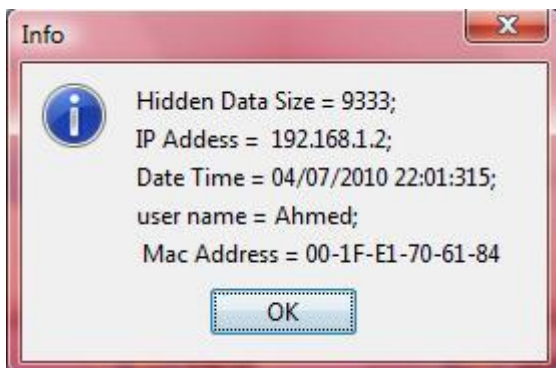


Figure 5.14 shows information about the sender.

If the size of the text is larger than needed by an image:

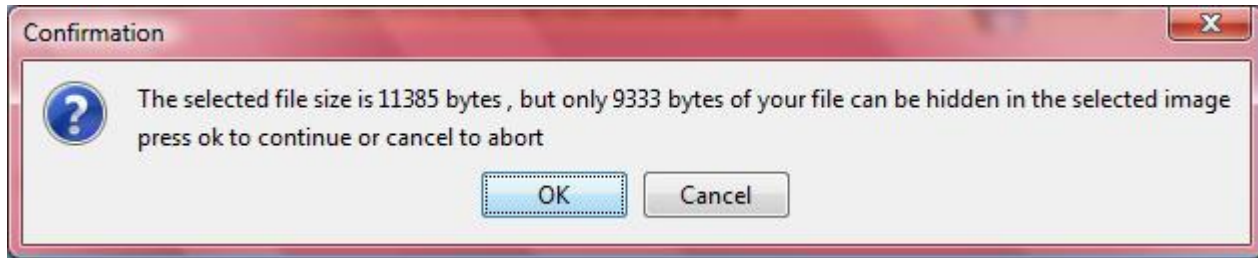
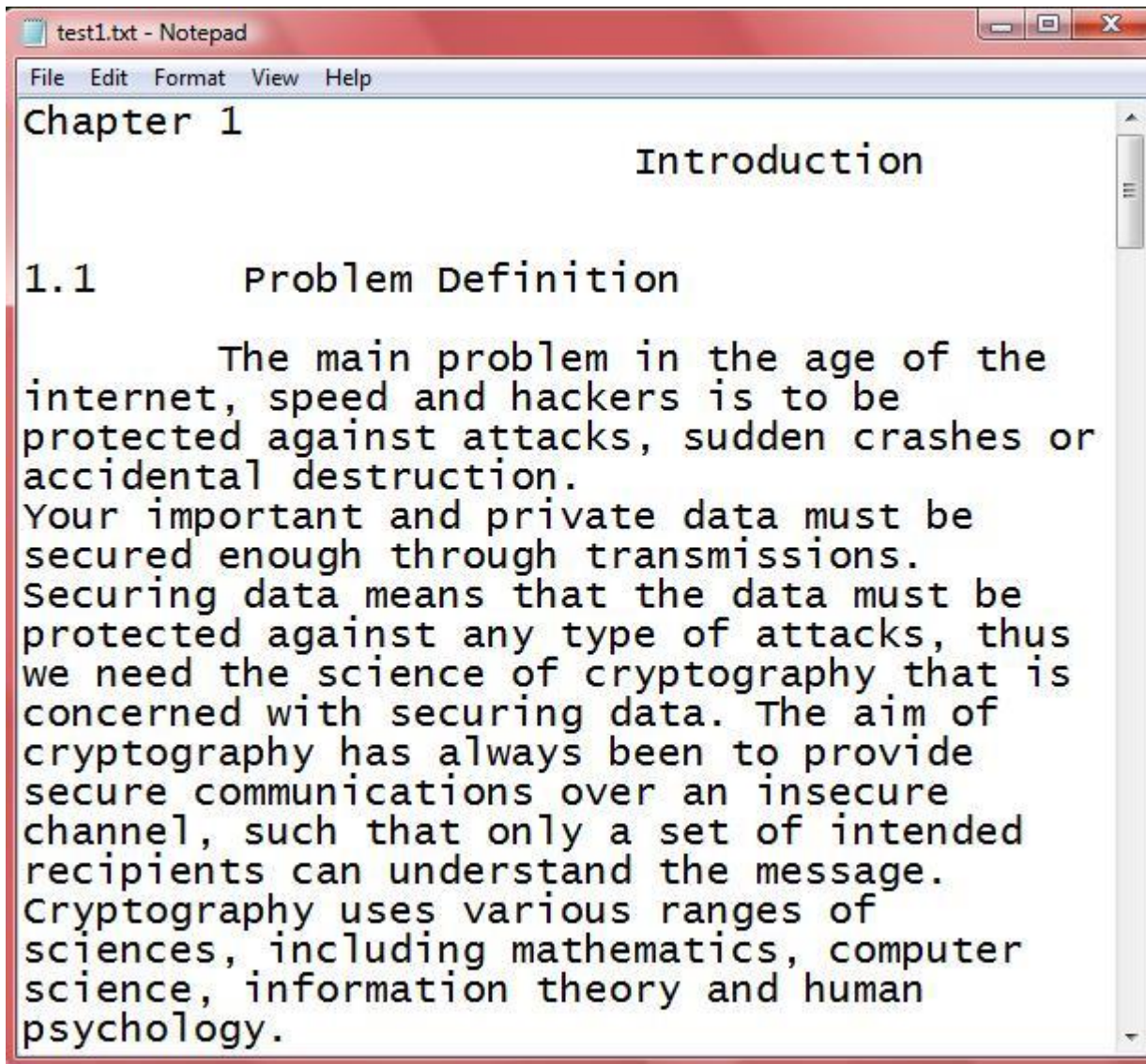


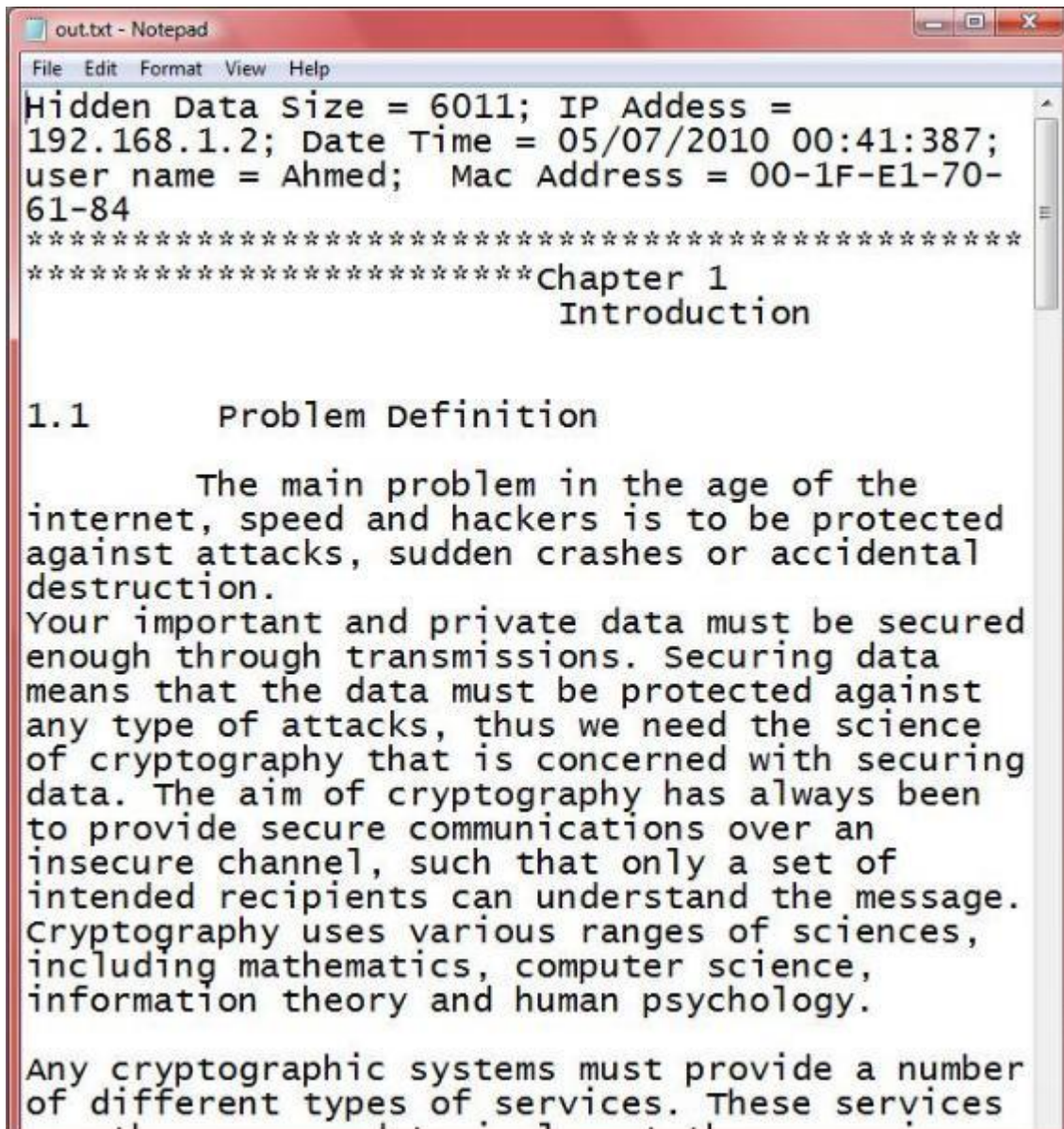
Figure 5.15 shows the suitable text file size for the input image.

Examples:

Text Before Hiding :



The extracted text file with the authentication message:



```
out.txt - Notepad
File Edit Format View Help
Hidden Data Size = 6011; IP Address =
192.168.1.2; Date Time = 05/07/2010 00:41:387;
user name = Ahmed; Mac Address = 00-1F-E1-70-
61-84
*****
*****Chapter 1
*****Introduction

1.1 Problem Definition

The main problem in the age of the
internet, speed and hackers is to be protected
against attacks, sudden crashes or accidental
destruction.
Your important and private data must be secured
enough through transmissions. Securing data
means that the data must be protected against
any type of attacks, thus we need the science
of cryptography that is concerned with securing
data. The aim of cryptography has always been
to provide secure communications over an
insecure channel, such that only a set of
intended recipients can understand the message.
Cryptography uses various ranges of sciences,
including mathematics, computer science,
information theory and human psychology.

Any cryptographic systems must provide a number
of different types of services. These services
```

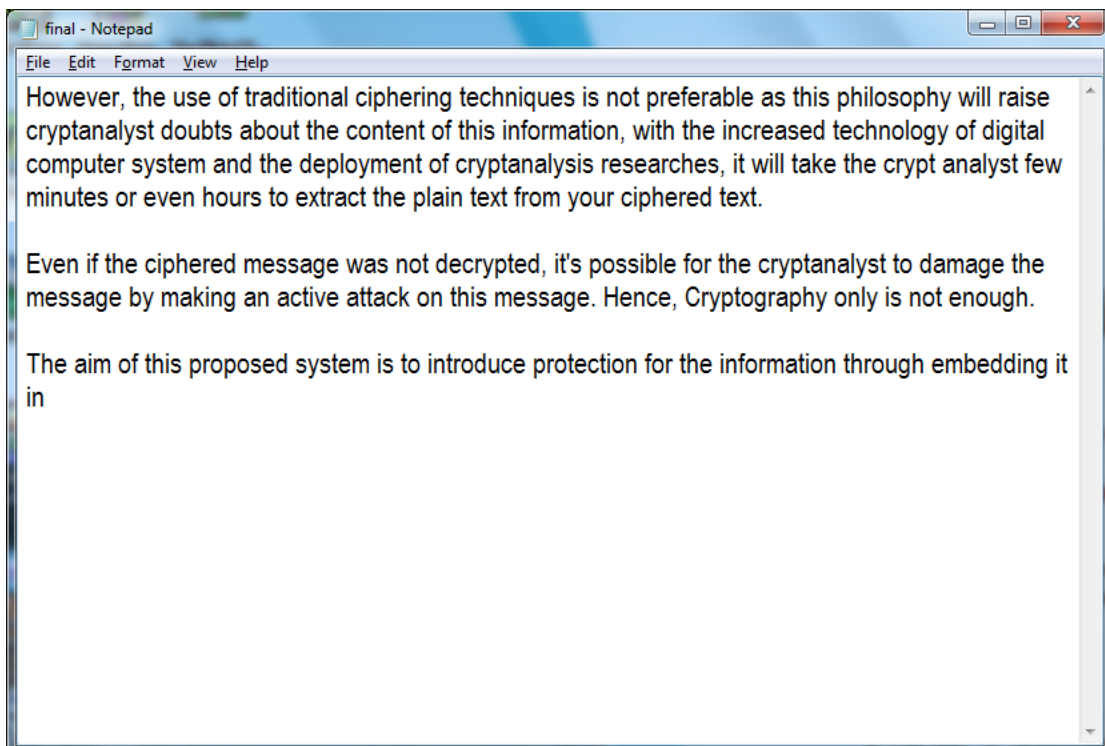
Different Image With other Size :



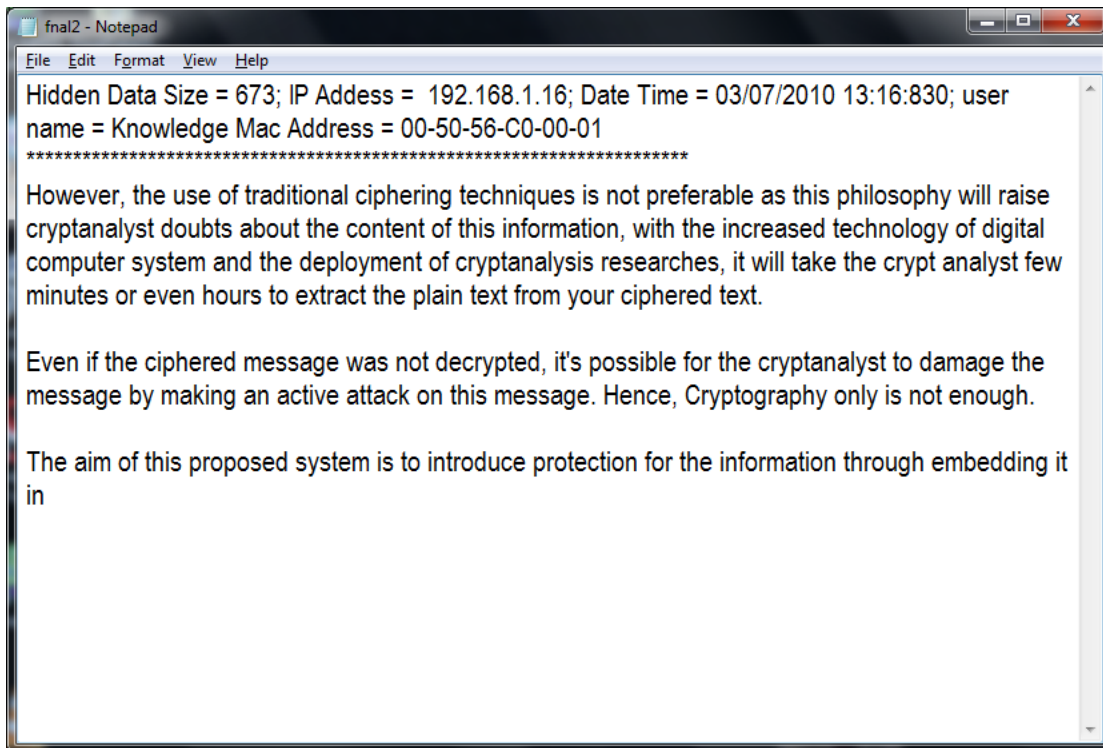
A: The Original Image

B: Image After Hiding

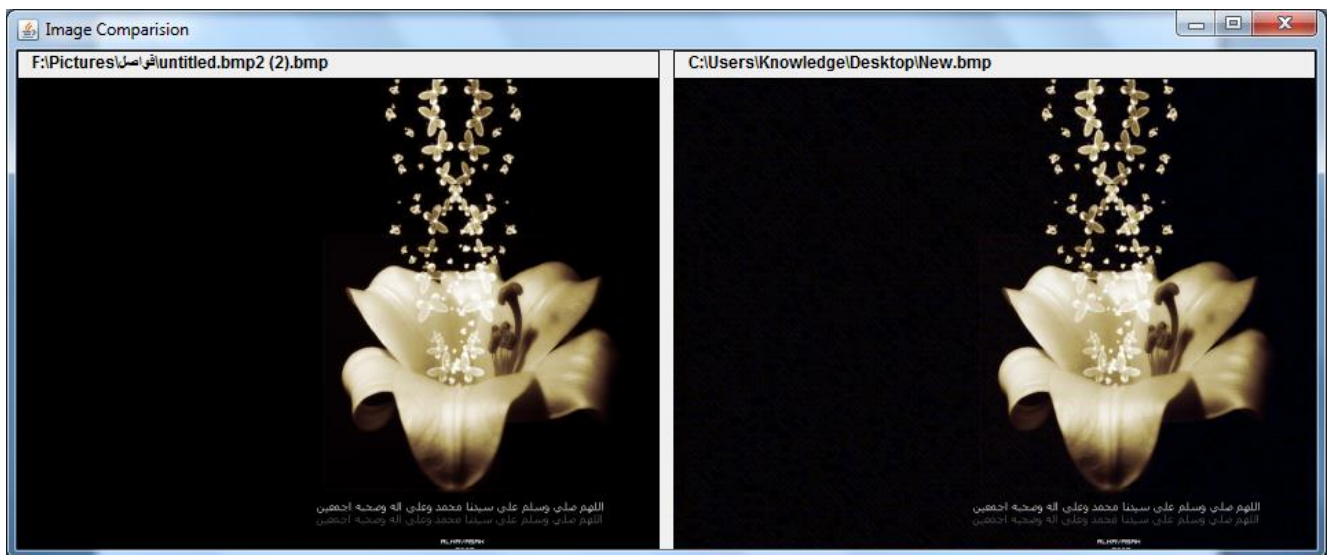
Text Before Hiding :



After Hiding :



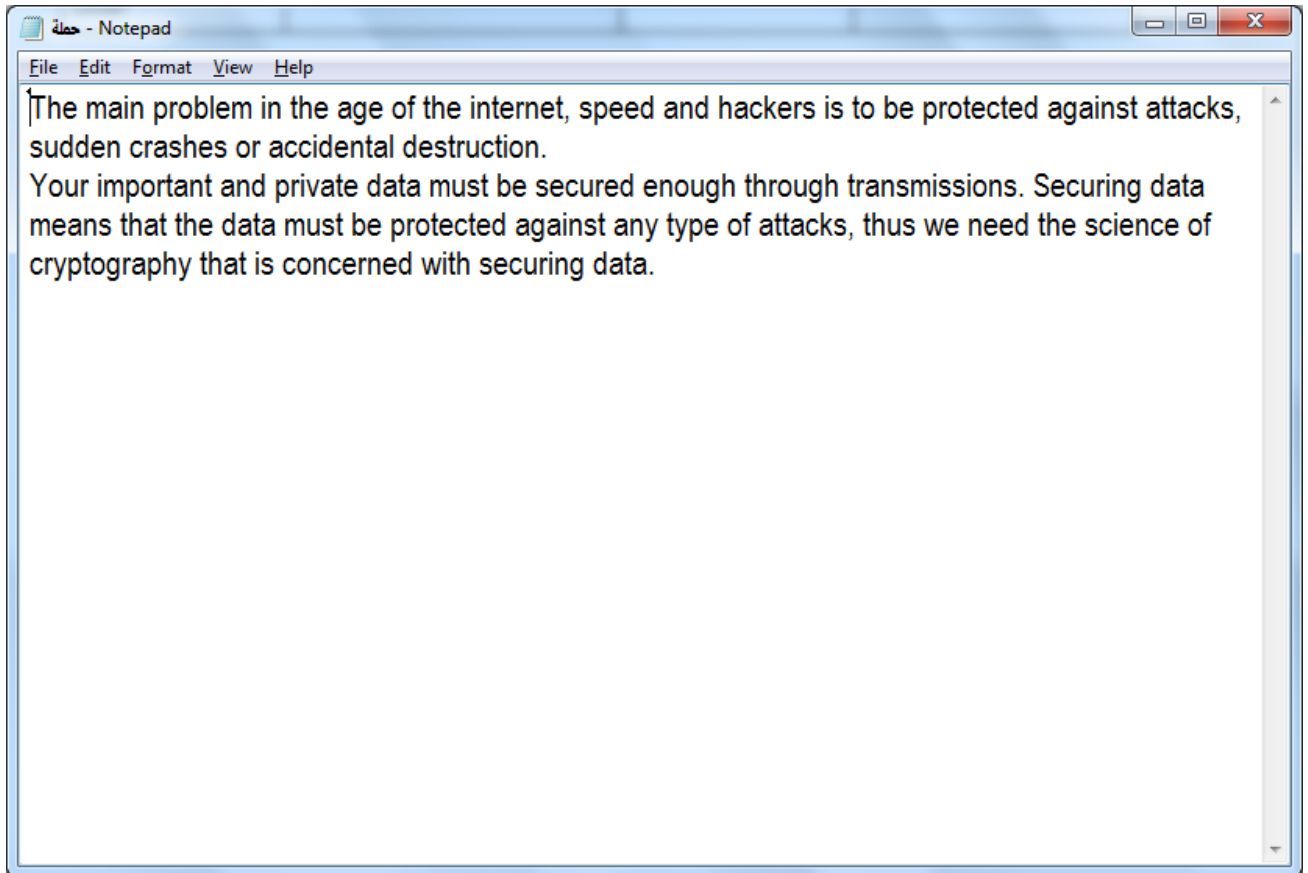
Another Different Image With Small Size :



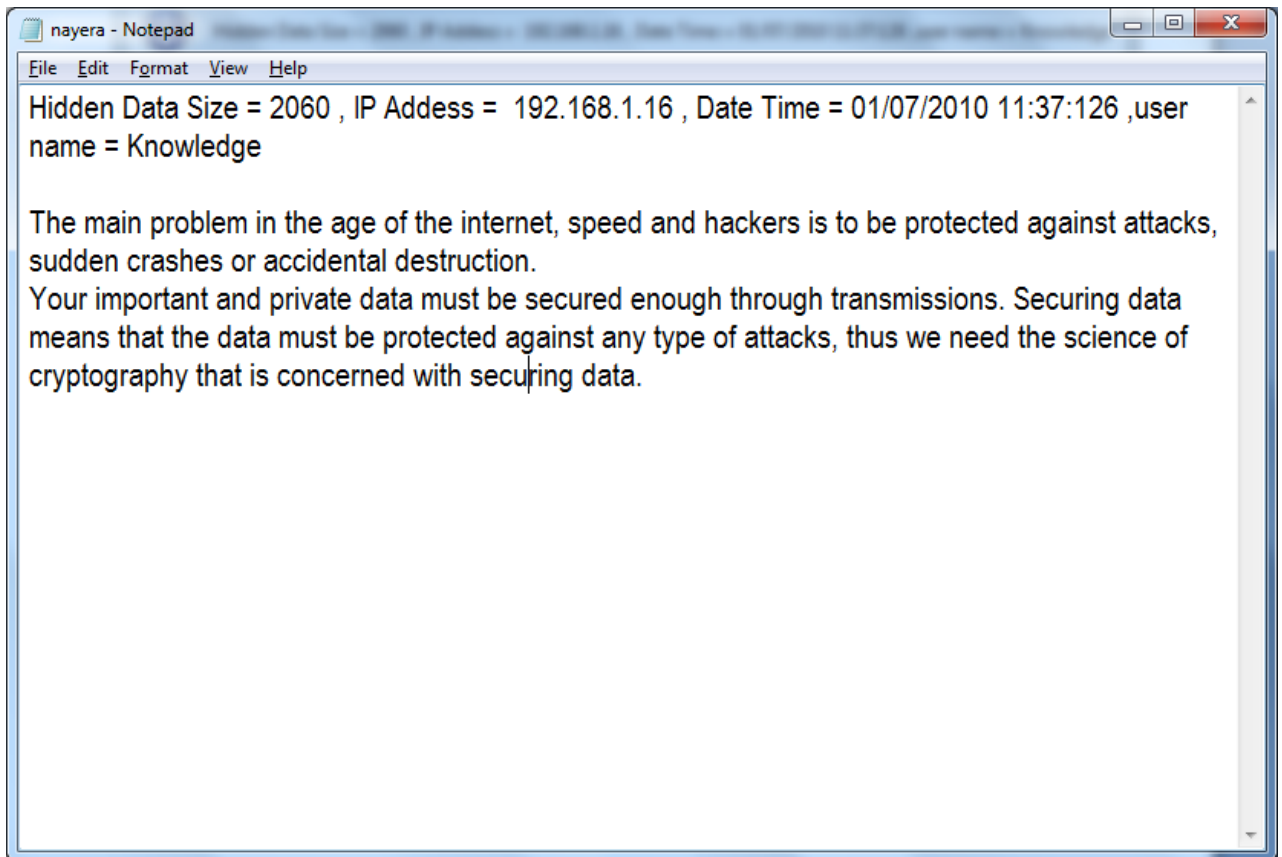
A : The Original Image

B : Image After Hiding

Text With Small Size Before Hiding :



Text with Small Size After Hiding :



References:

- [1] K. Lyons-Burke, Computer security, Federal Public Key Infrastructure Steering Committee, NIST Special Publication, October 2000.
- [2] Gonzalez, Digital Image Processing 2nd Edition, Prentice Hall, 2002.
- [3] Eric Hamilton, JPEG File Interchange Format Version 1.02 September 1, 1992.
- [4] Gregory K.Wallace, “The jpeg still picture compression standard”, IEEE 1991.
- [5] K. Ramchandran, A. Ortega, K. Metin Uz, and M. Vetterli, digital HDTV using joint “Multiresolution broadcast for source/channel coding.” IEEE Journal on Selected Areas in Communications, pp.6-23, January 1993.
- [6] M. W. Garrett and M. Vetterli, “Joint Source/Channel Coding Real Time Services on Packet of Statistically Multiplexed Networks,” IEEE/ACM Transactions on Networking, 1993.
- [7] W.Chu , DCT-based image watermarking using sub sampling, IEEE Transaction on Multimedia, pp.34–38, 2003.
- [8] R. J. Clark, “Transform Coding of Images,” New York: Academic Press, 1985.
- [9]Léger, A. Implementations of fast discrete cosine Global Telecommunications Conference, IEEE Communications Society, 1984.

- [10] N. Ahmed, T. Natarajan and K.R. Rao Discrete Cosine Transform,
IEEE Transactions on Computers, January 1974.
- [11] W.stallings, Cryptography and Network Security , 4th edition .
- [12] S.Smith, E.Richared, Authentication: From Passwords to Public Keys , Addison-Wesley Professional, October 1, 2001.
- [13] Barni M, Cox I. J, Kalker T, Kim H. J, Digital Watermarking, 4th International Workshop, IWDW, Proceedings Springer , 2005.
- [14] G. Langelaar, I. Setyawan, and R. Lagendijk,”Watermarking digital image and video data.”, IEEE Signal Processing Magazine, pp. 20–46, 2000.
- [15] J.R. HernMandez, M. Amado, F. PMerez-GonzMalez, “DCT-domain Watermarking techniques for still image detector performance Analysis and a new structure”, IEEE Trans, pp 55–68, 2000.
- [16] F.J. Macwilliams, N.J.A. Sloane, The Theory of Error-Correcting Codes, North-Holland, Amsterdam, 1996.
- [17] W.Chu , DCT-based image watermarking using sub sampling, IEEE Transaction on Multimedia, pp.34–38, 2003.
- [18] A. Manaf , M. Zeki,” Watermarking of Digital Images”, University Technology Malaysia / ATMA, 1st ENGAGE European Union - Southeast Asia ICT, Research Collaboration Conference, March 2006.
- [19] T.Addel Mageed , secure data hiding graduation project, Faculty Of Computers & Information – Cairo University ,2005.