A

Project Report on

(Image Encryption and Decryption Using AES & IWD)

Submitted in partial fulfillment of the requirements

for the award of the degree of

Bachelor of Technology

in

Information Technology

by

Neelesh Singh(1309713058),

Kunwar Digvijay Singh(1309713047),

Dinesh Kumar(1309713034)

Under the Supervision of

(Mr.Ranjeet Kumar)

Galgotias College of Engineering & Technology

Greater Noida 201306

Uttar Pradesh, INDIA

Affiliated to

Dr. APJ Abdul Kalam Technical University

Lucknow

(May 2017)

GALGOTIAS COLLEGE OF ENGINEERING & TECHNOLOGY
GREATERNOIDA - 201306, UTTAR PRADESH, INDIA.

CERTIFICATE

This is to certify that the project report entitled "**IMAGE ENCRYPTION AND DECRYPTION USING AES & IWD**" submitted by **Neelesh Singh(1309713058), Kunwar Digvijay Singh(1309713047), Dinesh Kumar(1309713034)** to the **AKTU** Uttar Pradesh in partial fulfillment for the award of Degree of Bachelor of Technology in Information Technology is a bonafide record of the project work carried out by them under my supervision during the year 2016-2017.

Dr. Bhawna Mallick
Professor and Head
Deptt. of IT

Name : **Ranjeet Kumar**
Designation: Asst. professor
Deptt. of IT

ACKNOWLEDGEMENT

This report is made under the guidance of Mr. Ranjit Kumar. We are highly obliged to the Head of the Department, Dr. Bhawna Mallick, for helping us and providing the valuable guidelines and healthy study environment. Firstly we would like to thank Mr. Ranjit Kumar, who helped us throughout the formation of this report and it is towards completion because of his valuable suggestions and guidelines. Then we would like to thank stackoverflow.com which guided us about the content of report and helped throughout the project report formation with its valuable suggestions and guidelines. Then at last but not least we would like to thank our parents and our dear friends for helping us in this project report and making it wonderful.

Neeelesh Singh
(1309713058)

Kunwar Digvijay Singh
(1309713047)

Dinesh Kumar
(1309713034)

ABSTRACT

Rapidly expanding networks are interconnecting more and more devices all over the globe, increasing both the number of interesting targets and the number of potential attackers. Moreover, eavesdropping on these networks has become much easier with the proliferation of wireless access points. Finally, the increasing complexity of communication and information systems makes their security much harder to control, giving rise to some rather unexpected new problems such as computer viruses and worms. This explains the importance of a strong interaction between cryptography, the field which studies techniques to protect information, and cryptanalysis, which focuses on methods to defeat this protection. There are so many cryptographic methods like AES, DES, TDES etc. But no method is capable to reduce this effect of attackers and can make the safer transactions. In this report, we will focus to enhance the methods of Advanced Encryption Standard (AES) by making it integrate with Intelligent Water Drops Algorithm. The proposed concept gives optimised results as compare to other Advanced Encryption Standard and Modified Advanced Encryption Standard.

Keywords— Advanced Encryption Standard, Intelligent Water Drops Algorithm, Cryptography, Image Security

CONTENT

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Many digital services require reliable security in storage and transmission of digital images. Due to the rapid growth of the internet in the digital world today, the security of digital images has become more important and attracted much attention. The prevalence of multimedia technology in our society has promoted digital images to play a more significant role than the traditional texts, which demand serious protection of users' privacy for all applications. Encryption and steganography techniques of digital images are very important and should be used to frustrate opponent attacks from unauthorized access.

Digital images are exchanged over various types of networks. It is often true that a large part of this information is either confidential or private. Encryption is the preferred technique for protecting the transmitted data. There are various encryption systems to encrypt and decrypt image data, however, it can be argued that there is no single encryption algorithm which satisfies the different image types .

In general, most of the available encryption algorithms are used for text data. However, due to large data size and real time constrains, algorithms that are good for textual data may not be suitable for multimedia data (Yas A. Alsultanny, 2008), (Droogenbroech and Benedett, 2002), (Fong and Singh, 2002). According to Xun (2001) and Wang (2005), even though triple-data encryption standard (T-DES) and international data encryption algorithm (IDEA) can achieve high security, they may not be suitable for multimedia applications (Xun et al, 2001), (Wang et al, 2005). Therefore, encryption algorithms such as data encryption standard (DES), advanced encryption standard (AES), and international data encryption algorithm (IDEA) were built for textual data (Lee et al, 2003), (Syed, 2002), (Xun et al, 2001).

Although we can use the traditional encryption algorithms to encrypt images directly, this may not be a good idea for two reasons. First, the image size is often larger than text. Consequently, the traditional encryption algorithms need a longer time to directly encrypt the image data. Second, the decrypted text must be equal to the original text but this requirement is not necessary for image data. According to Chang (2001), due to the characteristic of human perception, a decrypted image containing small distortion is usually acceptable (Chang et al, 2001), (Jiri Jan, 2005), (David Salomon, 2005). The intelligible information present in an image is due to the correlation among the image elements in a given arrangement. According to Mitra (2006), this perceivable information can be reduced

by decreasing the correlation among image elements using certain transformation techniques (Mitra et al, 2006).

In addition to cryptography, steganography techniques are getting significantly more sophisticated and have been widely used. The steganography techniques are the perfect supplement for encryption that allows a user to hide large amounts of information within an image. Thus, it is often used in conjunction with cryptography so that the information is doubly protected, that is, first it is encrypted, and then it is hidden so that an adversary has to find the hidden information before the decryption takes place.

1.1 Research Motivations

Most of the algorithms specifically designed to encrypt digital images were proposed in the mid-1990s. According to the Maniccam and Bourbakis (2004), there are two major groups of image encryption algorithms: (a) Non-chaos selective methods, and (b) Chaos-based selective or non-selective methods (Maniccam and Bourbakis, 2004). However, most of these algorithms are designed for a specific image format, either compressed or uncompressed. There are methods that offer light encryption (degradation), while others offer strong form of encryption. Some of the algorithms are scalable and have different modes ranging from degradation to strong encryption. According to Borko (2005), the user is expected to choose a method based on its properties, which will be best for image security (Borko Furht et al, 2005).

Image encryption has applications in internet communication, multimedia systems, medical and military imaging systems. Each type of multimedia data has its own characteristics such as high correlation among pixels and high redundancy. Thus, different techniques should be used to protect confidential image data from unauthorized access (Hossam El-din et al, 2006), (Ozturk and Sogukpinar, 2004).

The motivation behind this research is the ever-increasing need for harder-to-break encryption and decryption algorithms as the computer and network technologies evolve. We believe that by proposing the block-based encryption and decryption algorithm, it will help to reduce the relationship among image elements by increasing the entropy value of the encrypted images as well as lowering the correlation.

However, in order to increase the robustness of the proposed encryption technique, a steganography method using the least significant bit insertion will be applied without impacting the quality of the image.

Considering the above points, we will divide this research into three parts: a new transformation algorithm, a combination technique (transformation and encryption), and a

steganography approach that will be used to hide a secret information (that is, the number of horizontal and vertical blocks of the transformed image) in the encrypted image data before transmission to the receiver. A general block diagram of the transformation and encryption techniques is shown in Figure 1.1.

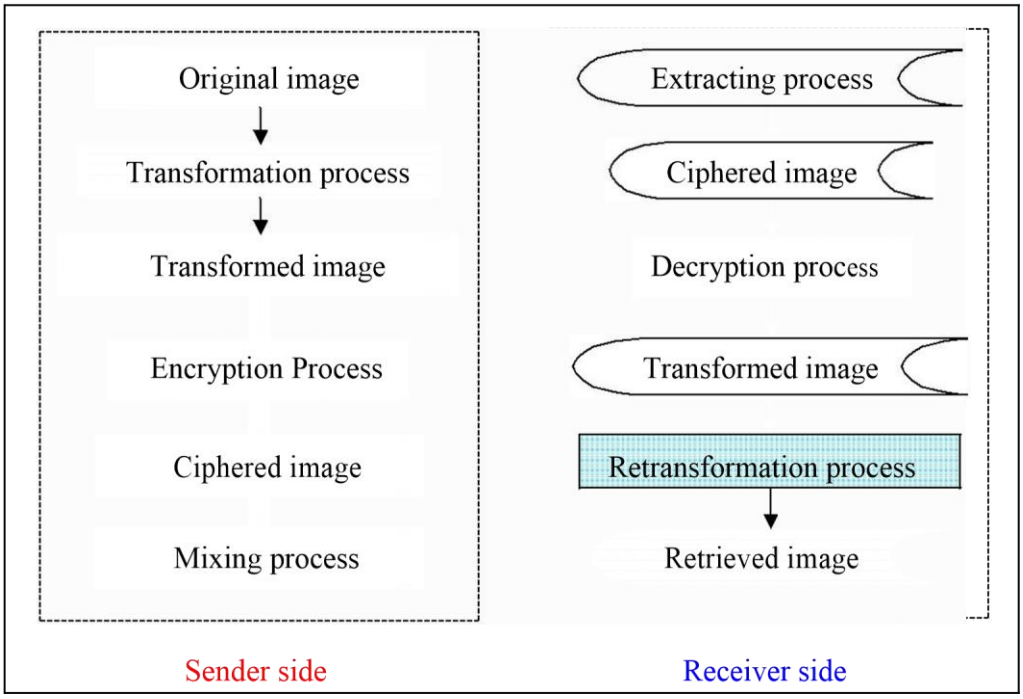


Figure 1.1 General block diagram of the proposed technique

1.2 Symmetric Key Algorithms

In general, symmetric key algorithms use a single, shared secret key. The same key is used for both encrypting and decrypting the data. There are two primary types of symmetric algorithms: block and stream ciphers. A block cipher is used to encrypt a text to produce a ciphertext, which transforms a fixed length of block data size into same length block of ciphertext in which a secret key and algorithm are applied to the block of data. For example, a block cipher might take a 64-bit block of plaintext as input, and output a corresponding 64-bit block of ciphertext. This transformation process should be conducted by a user providing a secret key and the decryption process is the inverse transformation to the ciphertext using the same key (April, 2005). Blowfish, Data Encryption Standard (DES), Triple-DES, IDEA, Rijndael and RC2 are examples of symmetric block cipher. The symmetric key algorithms use a single key for encryption and decryption processes as shown in Figure 1.2.

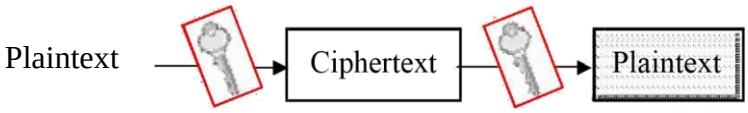


Figure 1.2 Symmetric key algorithms

The Blowfish algorithm is one of the symmetric block cipher algorithms that was designed in 1993 by Bruce Schneier as a fast alternative of the existing encryption algorithms, whereby it can be used as a replacement for the Data Encryption Standard (DES) or the International Data Encryption Algorithm (IDEA). The Blowfish encryption algorithm has been analyzed considerably, and is gaining acceptance as a strong encryption algorithm. Its source code is also available and it is not subjected to any patent royalties (Bruce Schneier, 1993), (William Stallings, 2003). Hence, this algorithm will be used mainly in this research as part of the new combination encryption technique.

The Blowfish algorithm consists of two parts: a key-expansion part and a dataencryption part. It encrypts the data by using the block cipher method, which breaks the text into 64-bit blocks before encrypting them. It takes a variable-length key from 32 bits to 448 bits of length, which implies flexibility in its security strength (John and James, 2005), (Bruce Schneier, 1993), (William Stallings, 2003).

1.3 Image Encryption Using Block-Based Transformation Algorithm

In this research, we propose a new transformation algorithm to be used as a preencryption transform, where the original image is divided into a random number of blocks which are shuffled and placed randomly within the image to build a newly transformed image. The generated transformed image is then fed to the Blowfish encryption algorithm. Thus, we expect that the combination of the transformation and encryption techniques will enhance the security level of the encrypted images.

This combination technique uses the original image to produce two output images:

a) a transformed image, using the proposed transformation algorithm.
b) a ciphered image of the transformed image, using the Blowfish algorithm.

A block diagram of the proposed technique versus Blowfish algorithm is shown in Figure 1.3.

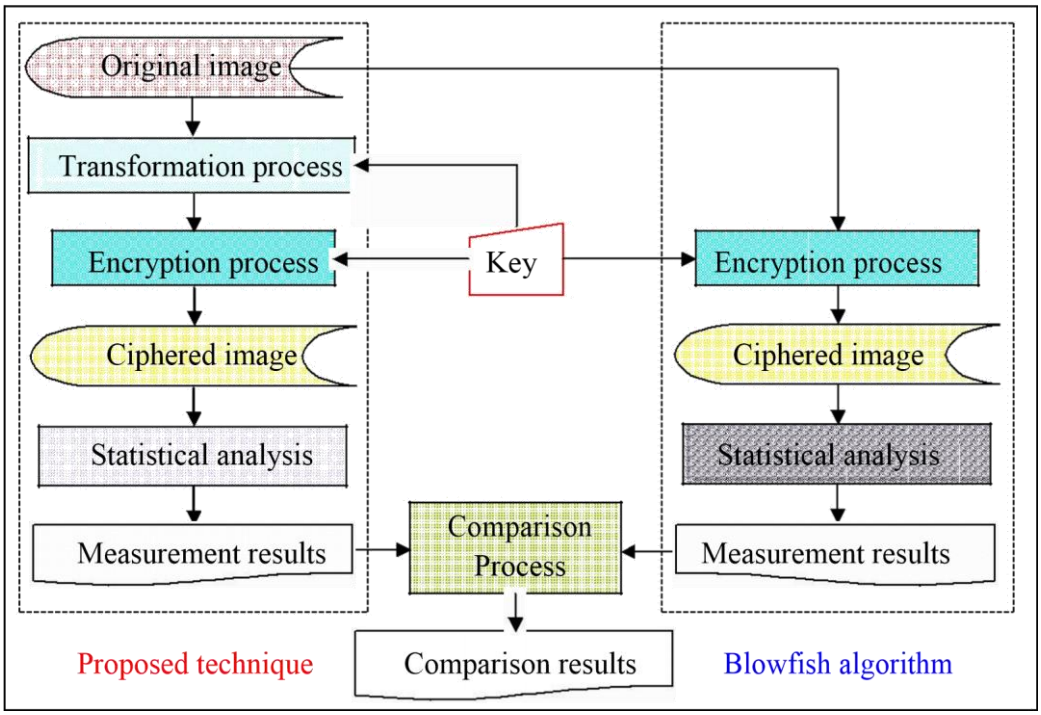


Figure 1.3 A block diagram of the proposed technique versus Blowfish algorithm

The measurements of correlation, entropy and histogram will be used to measure and compare the security level of the original image, transformed and encrypted images using the combination technique, and the encrypted image using the Blowfish algorithm alone. Encrypted images produced by the combination technique are expected to have lower correlation and higher entropy values, compared to those produced by the Blowfish algorithm alone.

In addition to transformation technique, we also present a steganography approach to be used for hiding secret information within encrypted image before being transmitted to the receiver.

1.4 Hiding Technique

Steganography techniques can be used for hiding information within other information. The least significant bit (LSB) insertion is one of the most widely used methods for embedding a message in a digital image. Steganography involves hiding information so it appears that no information is hidden at all. Therefore, it is expected that the person will not be able to decrypt the information (Shujun and Zheng, 2002), (Neil and Zoran, 2001), (Sushil, 2001). An alteration of the least significant bit of the color value of some pixels in an image will not change the quality of the image significantly. Therefore, a message can be sent within an image using these bits (Stallings, 2003).

In this research, the number of horizontal and vertical blocks of the transformed image, produced by the proposed algorithm, represents the secret information to be mixed (hidden) with the encrypted image before being transmitted to the receiver. This secret information will be needed at the receiver. Instead of sending the whole transformation table, which is usually big, only the secret information is sent. At the receiver side, the hidden information allows the receiver to regenerate the transformation table. Thus, the original image can be retrieved by the retransformation and decryption processes.

1.5 Goal, Scope, and Objectives of the Research

The goal of this research is to enhance the security level of the encrypted images using the proposed transformation algorithm. The scope is limited to the image encryption using the combination technique (block-based transformation algorithm and Blowfish encryption algorithm) on Microsoft windows based machine. This combination technique is applied to divide and shuffle the positions of the blocks of the original image, encrypt the transformed image, and then embed secret information (the number of horizontal and vertical blocks) in the encrypted image data prior to transmission to the receiver. Furthermore, the focus of this research was concerning a bit mapped (bmp) images using the standard Cipher Block Chaining (CBC) mode of the Blowfish algorithm.

To achieve the above goal, the objectives of this research will be as follows:

1. To introduce a new algorithm for image transformation, and to test and evaluate it.
2. To compute and compare correlation, entropy and histogram of different images with and without the proposed algorithm. To compare the security levels of the encrypted images generated by the combination technique and the Blowfish algorithm..
3. To introduce a steganography method to exchange the secret information between the sender and the receiver that will be used for producing the transformation table.

1.6 Thesis Structure

This thesis will be organized into six chapters. Chapter 1 provides an introduction to the work, the motivations of this research, and explains important goal, scope, and objectives of this study. Image encryption using block based transformation algorithm is explained to provide a general aspects of symmetric encryption algorithm, then the most important features of the blowfish encryption and decryption algorithm are presented. A precise description of the proposed technique and its diagram is presented. Furthermore, the important use of the steganography technique is also presented.

The general aspects of digital images and image file formats will also be discussed. We will also provide the explanations on cryptographic systems in general; block cipher with its modes of operation, stream cipher and some of the most commonly used or well known encryption and decryption algorithms such as DES, Blowfish, Rijndael-AES and IDEA. This chapter also highlights a background of the current research in image encryption. Steganography technique for digital images is also presented. At the end of this chapter, the image measurements; the correlation among image elements, image entropy, and image histogram as well as image similarity are discussed in certain degree of details.

In chapter 3, we will present and discuss in details the description of the model and methodology that will be applied to enhance the security level of the encrypted images by using the newly proposed approach. Hiding information in image using steganography technique is also presented in this chapter.

In chapter 4, we will discuss the implementation, testing and results analysis of the proposed technique. Hiding efficiency will also be presented in this chapter with the focus on mixing and extracting data within the encrypted image. How the sender will hide the secret information in the image data that enables the receiver to rebuild the transformation table using the secret key will also be discussed.

In chapter 5, we will present the technical contribution; strength and efficiency of the technique by comparing the combination technique with three commonly used encryption algorithms; Blowfish, Twofish, or RijnDael. Economical contribution of the research; knowledge gain and security enhancement will also be explained. Some suggestions for future work will also be provided. Chapter 6 will summarize the main points of the thesis and list the main conclusions of the work.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter provides a detail description of the cryptographic systems used in this project Section 2.2 presents general aspects of the digital image encryption. Section 2.3 defines fundamental concepts of cryptography systems and image encryption. Categories of cryptography systems are also discussed; symmetric key cryptography and public key cryptography. Some important symmetric key algorithms such as block cipher and stream cipher algorithms are also introduced in this section. Furthermore, we will discuss modes of operation; Electronic Code Book Cipher (ECB), Cipher Block Chaining (CBC), Cipher Feedback (CFB), and Output Feedback (OFB).

2.2 Digital Images

A digital image is defined by 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 (Steinmetz and Nahrstedt, 2002), (Kristian Sandberg, 2000).

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 (David Salomon, 2007).

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) that represents the color for each value. The color space is computed using the following
equation:

ColorSpace = 2^b…….....Equation 1

where: b: 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 (Vaughan, 2004), (Rafael and Richard, 2002), (Sander, 2000). Table 2.1, illustrates the image color space.

Table 2.1 Image color space versus bit depth (bpp)

Image properties	Bits resolution	Color space
Binary image (black and white)	1	2 colors
Gray scale (monochrome)	8	256 gray levels
Colored image	8	256 colors
Colored image	16	65536 colors
True color (RGB)	24	16,777,216 colors

As seen from Table 2.1, 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 × BR ……..……..Equation 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 × H × BR = 640 × 480 × 24 bits = (7372800/1024/8) = 900 KB.

2.2.1 Digital Image Formats

Basically, there are three types of image files: bitmap, vector, and metafiles. When an image is stored as a bitmap file, its information is stored as a collection of pixels, manifest as colored or black-and-white dots. When an image is stored as a vector file, its information is stored as mathematical data. The metafile format can store image information as pixels (bitmap), mathematical data (vector), or both (Betcher and Gardner, 2006), (Sander, 2000). There is no single format that is appropriate for all types of images. According to Glouglim (2001), larger files take longer to load, require more disk space and can take longer to print, whereas small file sizes means greater performance (McGlouglim, 2001). The most common file formats are discussed below:

a) BMP Files

According to Bourke (1998), the BMP Bitmaps are defined as a regular rectangular mesh of cells called pixels (Bourke, 1998). Each pixel contains a color value as shown in Figure 2.1

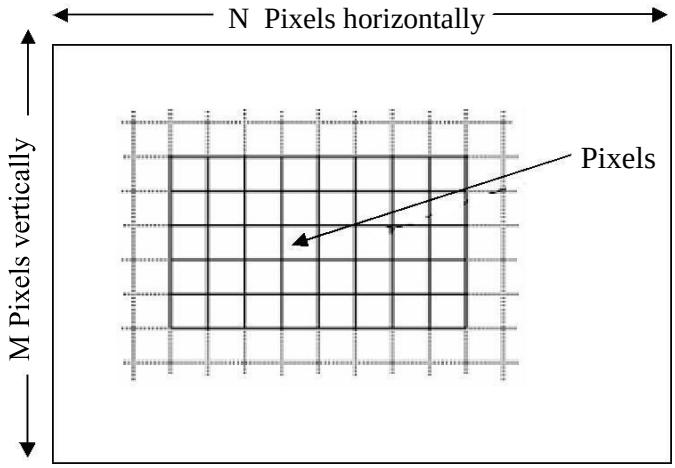


Figure 2.1 Image pixels

Bitmaps are characterized by only two parameters: the number of pixels, and the information content (color depth) per pixel, and they are the most commonly used type to represent images on the computer.

BMP is the native bitmap format of Windows. BMP is a general format that stores images in different color depths without compression (Betcher and Gardner, 2006), (Paul Bourke, 1998), (James and William, 1996).

The BMP advantages are that each pixel is independently available for any alteration or modification, and that repeated use of them does not normally degrade the image quality (Lancaster, 2003). The main disadvantage of this format is due to the size of the files, which is usually larger compared to other formats or other lossy compression schemes. In general, a BMP file consists of a header, descriptive information about the image (such as width, height, etc), an optional Color Lookup Table (CLUT) area which contains the actual colors of the image pixels, and a pixel data area (Shalini, 2004). The parts of the BMP file are illustrated in Table 2.2.

Table 2.2 BMP file header

Header	Stores general information about the BMP file.
Information header	Stores detailed information about the bitmap image.
Optional palette	Stores the definition of the colors being used for indexed color bitmaps.
Image data	Stores the actual image, pixel by pixel.

Color Lookup Tables (CLUTs) follow the header for those lower performance modes in which they are used such as (1, 4, 8 or 16 lookup colors). These, in turn, are followed by the actual pixel data. The main interest lies in the 24-bit uncompressed RGB color mode. In this mode, there are no color lookup tables used. Each pixel consists of an 8-bit red value, an 8-bit green value and an 8-bit blue value (Lancaster, 2003).

B) GIF Files

The Graphics Interchange Format (GIF) was originally developed by CompuServe in 1987. It is one of the most popular file formats for web graphics and exchanging graphics files between computers. The GIF format supports 8 bits of color information that is limited to 8 bits palette and 256 colors. Thus, only 256 different colors are available to represent the picture. It can be viewed by all common browsers. GIF also support animation, transparency and interlacing (Betcher and Gardner, 2006), (Robert Fry, 2006).

GIF images are automatically compressed when they are saved using a lossless compression method known as LZW (Lempel-Ziv-Welch) that does not degrade the image quality. GIF format provides four main features: interlacing, transparency, file compression, and primitive animation. The interlacing feature allows the browser to display portions of the image as it updates. The original image starts off with poor quality but gets better as more

of the interlacing parts are updated. Interlaced GIF files allow users to view a portion of the image as the file is loading (Seeram and Radiography, 2006)

One of GIF's weaknesses is that GIF images are limited to a maximum of 256 colors. The quality of the image suffers if the color depth is reduced to less than the color depth of the original image. GIF files can store any of the 16.8 million colors but only a maximum 256 colors in each GIF file. Therefore, when converting an image to GIF, the program compresses the file by reducing the number of colors in the image from 24-bit (millions of colors) to 8-bit (256 colors). However, the GIF file format has the ability to store multiple images in a single file and play the images in a loop, thereby giving the appearance of animation (CIMC, 2006), (Sharon Wheeler,2000).

c) JPEG Files

The Joint Photographic Experts Group, (JPEG) format, is one of the most popular formats for web graphics. It supports 24 bits of color information. The JPEG file format stores all of the color information in an RGB image, and then it compresses the file size to save storage space, or it saves only the color information that is essential to the image. Unlike GIF, JPEG does not support transparency.

The compression method used in JPEG is usually lossy compression, meaning that some visual quality is lost in the process. JPEGs can be saved in a variety of lossy compression levels. This means more or less compression can be applied to the image, depending upon which looks best. JPEG can be used by almost any browser. Since JPEG is an image compressor, it is best used for photographic quality images and detailed illustrations with many colors (Tom Lane, 2008).

The advantage of JPEG is that it is a highly compressed file format. Therefore, the image can be compressed while the quality is maintained. JPEG weakness is that lossy compression may result in low quality graphics. Another weakness noted in JPEG formats is that there is no support for pixel transparency. JPEGs lose quality every time they are opened, edited and saved. It is very important to minimize the number of editing sessions between the initial and final version of a JPEG image (Sharon Wheeler, 2000), (CIMC, 2006), (Graphics Academy, 1998).

d) PICT Files

The Picture File Format (PICT) is used primarily on the Macintosh platform. It is the default format for Macintosh image files as its standard metafile format. The PICT format is most commonly used for bitmap images, but can be used for vector images as well. The PICT is a lossless format. Since the PICT format supports only limited compression on Macintoshes

with QuickTime installed, PICT files are usually large. PICT is used for images in video editing, animations, desktop computer presentations, and multimedia authoring (Chris Betcher and Margie Gardner, 2007).

e) EPS Files

The Encapsulated PostScript (EPS) file format is intended to make files usable as a graphics file format. The EPS file format is a metafile format. It can be used for vector images or bitmap images. It can also be used on a variety of platforms, including Macintosh and Windows. If an EPS image is placed into a document, we can scale it up or down without information loss (Chris Betcher and Margie Gardner, 2007).

f) PNG Files

The Portable Network Graphics (PNG) format is a bitmapped image format that employs lossless data compression. It will likely be the successor to the GIF file format. PNG is expected to become a mainstream format for web images and could replace GIF entirely. It is platform independent and should be used for single images only (not animations). Compared with GIF, PNG offers greater color support and better compression, gamma correction for brightness control across platforms, better support for transparency, and a better method for displaying progressive images (Sharon Wheeler, 2000), (Fulton, 2005).

g) TIFF Files

The Tag Interchange File Format (TIFF) is a tag-based international standard for storing and interchanging bitmaps between applications and hardware platforms. It is compatible with a wide range of software applications and can be used across platforms such as Macintosh, Windows, and UNIX. The TIFF format is complex, thus TIFF files are generally larger than GIF or JPEG files. TIFF supports lossless LZW compression. However, compressed TIFF takes longer to open. The format consists of items called tags which are defined by the standard. Each tag is followed by a tag dependent data structure (Graphics Academy, 1998). The next section will explain the cryptographic system and image encryption.

2.3 Encryption algorithms

Cryptography is the conversion of data into a secret code for transmission over a public network. Cryptography enables the sender to securely store sensitive information or transmit it across insecure networks so that it cannot be read by anyone except the intended recipient (Harris Chen, 2001), (Gary Kessler, 2007).

Encryption of sensitive data is necessary. Cryptography is used to render the information unintelligible if transmission is intercepted by unauthorized individuals (Jae Shim, 2000). The intelligible form (original data) of information is called plaintext and the unintelligible form (protected data) is called ciphertext (Elbirt and Paar, 2005), (Stallings, 2003). The process of converting the plaintext into ciphertext is called encryption, while the reverse process of transforming ciphertext into the corresponding plaintext is called decryption.

In general, most cryptographic algorithms use a secret value called a key. The security of encrypted data is entirely dependent on two things: the strength of the cryptographic algorithm and the secrecy of the key. The key is used for encryption and decryption and must be kept secret, thereby requiring the sender and receiver to agree on the same key before making any data transmissions. The key is independent of the plaintext. Therefore, the same plaintext encrypts to different ciphertext with different keys, and thus both processes are impossible without the use of the correct key (Weber and Fahrny, 2003), (Natasa, 2005), (Schneier, 1996).

2.3.1 Data Encryption Standard (DES) Algorithm

Cryptography can be strong or weak. Cryptographic strength is measured in the time and resources it would require to recover the plaintext. The result of strong cryptography is ciphertext that is very difficult to decipher without possession of the appropriate decoding tool. The sender and the recipient must keep the key secret because anyone who knows the key can use it to decrypt the plaintext. In addition, the strength of the algorithm is important. An unauthorized entity can take encrypted ciphertext and attempt to break the encryption by determining the key based on the ciphertext (Zhong et al, 2005).

While cryptography is the science of securing data, cryptanalysis is the science of analyzing and breaking secure communication, and therefore it is the process of recovering the plaintext or key, usually by using the ciphertext and knowledge of the algorithm (Albassal and Wahdan, 2004), (Bagnall et al, 1997), (Harris Chen, 2001). According to Kessler (2007), there are two types of algorithms: symmetric that uses a single key for both encryption and decryption, and asymmetric that uses one key for encryption and another for decryption (Kessler, 2007).

Cryptography can also be used to ensure the security of the communication path through the following: (a) data integrity which means ensuring that the data has not been modified by unauthorized entities. Thus, the message received by the recipient is the same as the message sent by the sender. (b) Non-repudiation ensures that the sender of any message cannot deny his/her actions. This can be achieved with digital signatures in conjunction with

asymmetric key encryption. (c) Authentication is the process of proving the identity, and (d) privacy/confidentiality is the process to ensure that no one can read the message except the intended recipient (Alina Stan, 2007), (Solomon and Chapple, 2005).

The modern field of cryptography can be divided into several areas of study. In this section, two related categories for cryptography systems: public key cryptography and symmetric key cryptography will be discussed (Kaufman et al, 2002).

Public Key Cryptography (PKC) is also known as asymmetric cryptography. It uses one key for encryption and another for decryption. The encryption key known as public key is intelligible and can be distributed for all parities, while the decryption key known as private key is intelligible only to the recipient. Each user creates a pair of keys; if one is used for encryption then the other is used for decryption (Kaufman et al, 2002). The public and private keys are mathematically related so that data encrypted with the public key can only be decrypted with the private key. This guarantees message privacy during transit. An important characteristic of public key encryption algorithms is that it should be computationally infeasible to determine the decryption key given only knowledge of the algorithm and the encryption key. Public key encryption can be used to exchange the secret key between the parties in a symmetric key cryptosystem (Stallings, 2003), (Baek, 2004). Figure 2.2, shows the main ingredients of public key cryptography system.

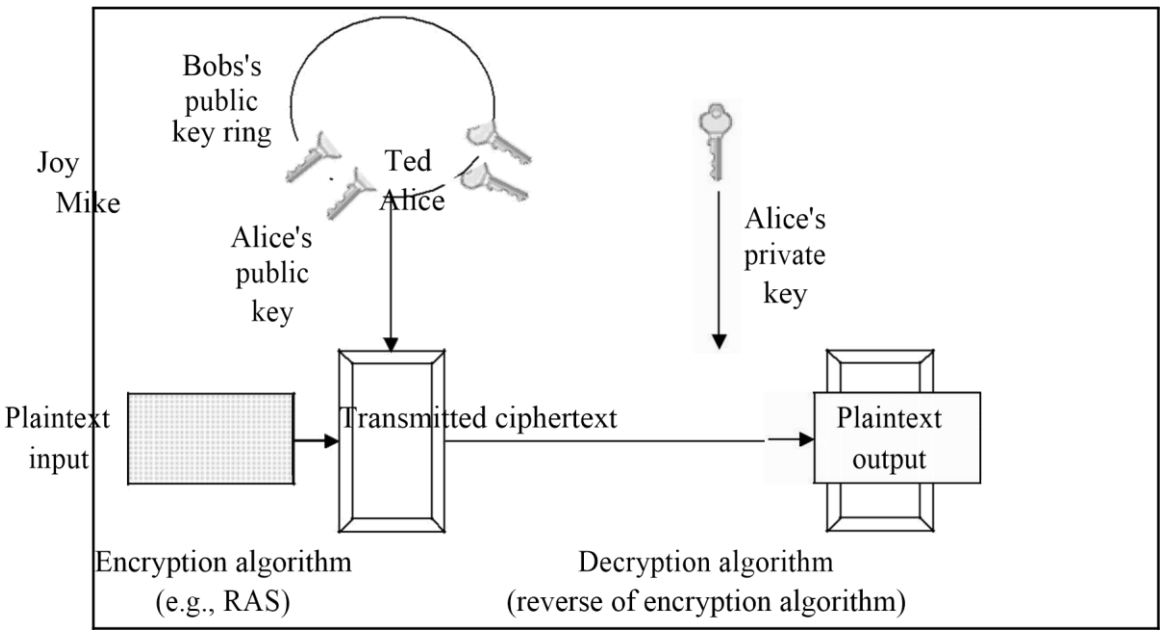


Figure 2.2 Public key encryption and decryption model (Stallings, 2003).

As illustrated in Figure 2.2, there are three basic steps to send a message by using public key encryption:

- Sender and receiver exchange the public keys, while the private key is kept secret by its owner.
- The sender uses the recipient's public key in encrypting a message for sending.
- The recipient's secret key is used to decrypt the received message.

In symmetric key cryptography, encryption and decryption are performed using the same secret key.

The key can only be known by the sender and receiver to maintain integrity (Thomas Shinder, 2002). According to Whitman and Jason (2005), the primary disadvantage of symmetric key algorithms is that the key must remain secret at all times. For this reason, the key must be protected and secured requiring the sender to transmit the key to the recipient in a secure fashion (Jason Isom, 2005), (Whitman and Mattord HJ, 2005). Symmetric encryption is the most widely used algorithm.

2.3.2 Advanced Encryption Standard (AES) Algorithm

For Rijndael, the length of both the block to be encrypted and the encryption key are not fixed. They can be independently specified to 128, 192 or 256 bits. The number of rounds, however, varies according to the key length. It can be equal to 10, 12 and 14 when the key length is 128bits, 192 bits and 256 bits, respectively. The basic components of Rijndael are simple mathematical, logical, and table lookup operations. The latter is actually a composite function of an inversion over Galois Field (GF) with an affine mapping. Such structure makes Rijndael suitable for hardware implementation.

The algorithm is based on AES Key Expansion technique.

AES Key Expansion technique in detail.

AES Key Expansion Pseudo code for AES Key Expansion: The key-expansion routine creates round keys word by word, where a word is an array of four bytes. The routine creates 4x(Nr+1) words. Where Nr is the number of rounds. The process is as follows

The first four words are made from the cipher key (initial key). The key is considered as an array of 16 bytes (k0 to k15). The first four bytes (k0 to k3) become w0, the four bytes (k4 to k7) become w1, and so on.

The rest of the words (wi for i=4 to 43) are made as follows If (i mod 4) != 0, wi=wi-1 xor wi4. If (i mod 4) = 0, wi=t xor wi-4. Here t is a temporary word result of applying SubByte transformation and rotate word on wi-1 and XORing the result with a round constant.

1Modifications in AES Key Expansion Certain changes made in the above key expansion process improves the encryption quality, and also increases the avalanche effect. The changes are a) The Rcon value is not constant instead it is being formed from the initial key itself, this improves the avalanche effect. b) Both the s-box and Inverse s-box are used for the Key Expansion process which improves nonlinearity in the expanded key and also improves the encryption quality. c) We do not use the S-box and Inverse S-box as such for this algorithm; instead we perform some circular shift on the boxes based on the initial key this improves the key sensitivity.

The above changes in the algorithm can be represented as

1) Formation of Rcon values Rcon [0]=key[12:15]; Rcon [1]=key[4:7]; Rcon [2]=key[0:3]; Rcon [3]=key[8:11];

2) Using Inverse S-Box for key expansion The 'temp' value used in the algorithm is formed as temp = SubWord(RotWord(temp)) XOR InvSubWord(Rcon[i/4]); Where InvSubWord:
InverseSubByte transformation table value

2. Steps Involved 2.3.1 Key Selection The sender and receiver agree upon a 128 bit key. This key is used for encryption and decryption of images. It is a symmetric key encryption technique, so they must share this key in a secure manner. The key is represented as blocks k[0],k[1]...k[15]. Where each block is 8bits long (8*16=128 bits).

3. Generation of Multiple keys The sender and receiver can now independently generate the keys required for the process using the above explained Modified AES Key Expansion technique. This is a one time process; these expanded keys can be used for future communications any number of times till they change their initial key value.

4. Encryption Encryption is done in spans, where we process 16 pixels in each span. For both its Cipher and Inverse Cipher, the AES algorithm uses a round function that is

composed of four different byteoriented transformations: SubBytes, ShiftRows, MixColumns and AddRoundKey.

5. Decryption The decryption process is similar as encryption, but we use Inverse SubByte Transformation.

2.3.3 Serpent Algorithm

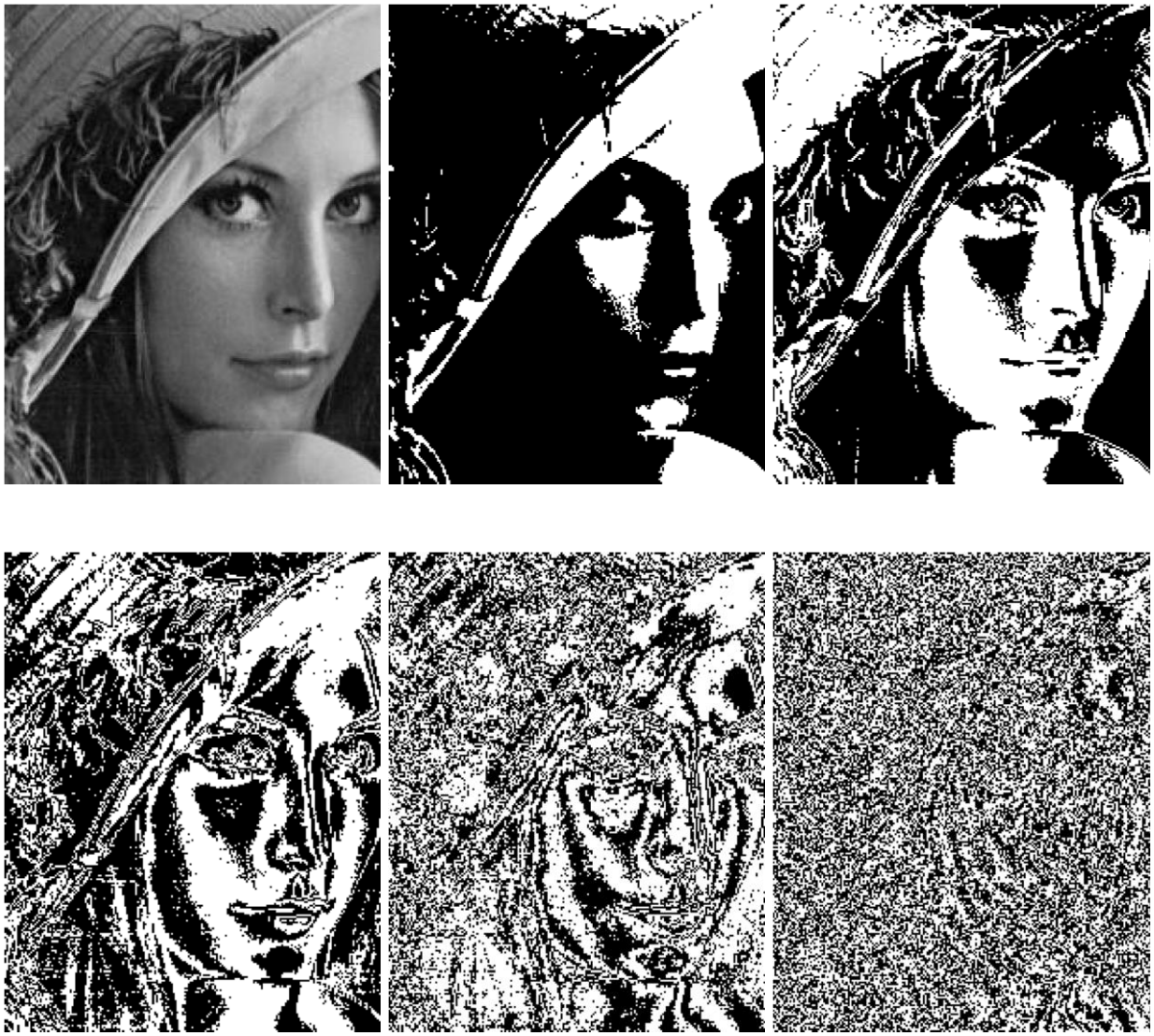
Like other AES submissions, Serpent has a block size of 128 bits and supports a key size of 128, 192 or 256 bits.The cipher is a 32-round substitution-permutation network operating on a block of four 32-bit words. Each round applies one of eight 4-bit to 4-bit S-boxes 32 times in parallel. Serpent was designed so that all operations can be executed in parallel, using 32 bit slices. This maximizes parallelism, but also allows use of the extensive cryptanalysis work performed on DES.

Serpent took a conservative approach to security, opting for a large security margin: the designers deemed 16 rounds to be sufficient against known types of attack, but specified 32 rounds as insurance against future discoveries in cryptanalysis. The official NIST report on AES competition classified Serpent as having a high security margin along with MARS and Twofish, in contrast to the adequate security margin of RC6 and Rijndael (currently AES). In final voting, Serpent had the least number of negative votes among the finalists, but scored second place overall because Rijndael had substantially more positive votes, the deciding factor being that Rijndael allowed for a far more efficient software implementation.

The Serpent cipher algorithm is in the public domain and has not been patented. The reference code is public domain software and the optimized code is under GPL. There are no restrictions or encumbrances whatsoever regarding its use. As a result, anyone is free to incorporate Serpent in their software (or hardware implementations) without paying license fees.

2.4 Current Research in Image Encryption

2.4.1 Techniques for a Selective Encryption of Uncompressed and Compressed Images:



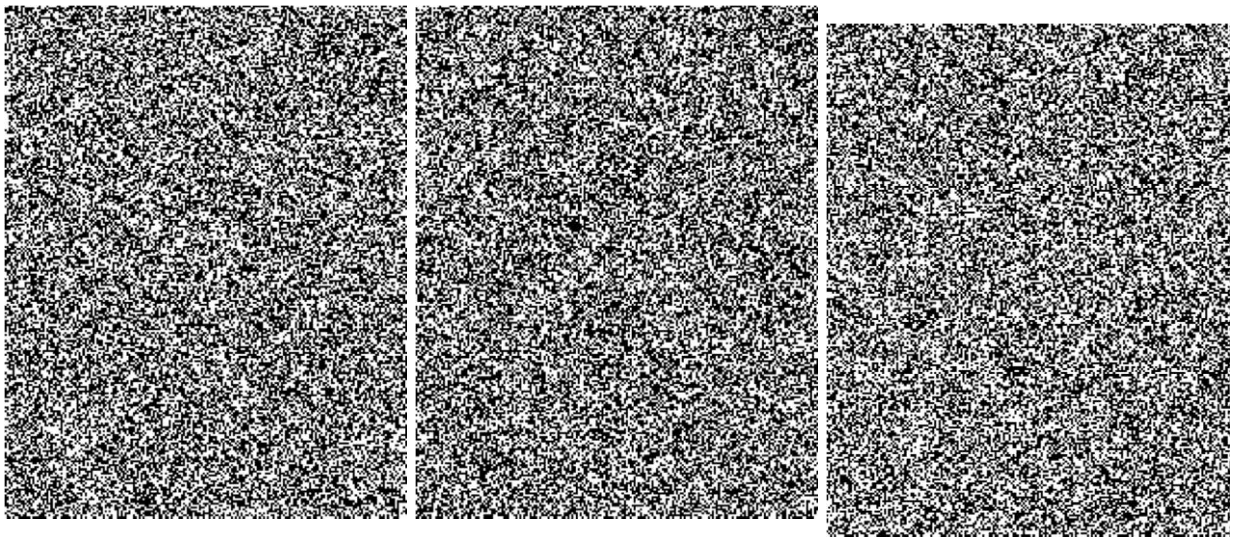


Figure 2.3: LENA and her bitplanes (i_7, ..., i_0) starting from the most significant bit.

In some applications, it is relevant to hide the content of a message when it enters an insecure channel. The initial message prepared by the sender is then converted into ciphertext prior to transmission. The process of converting plaintext into ciphertext is called encryption (for a review on encryption techniques). The encryption process requires an encryption algorithm and a key. The process of recovering plaintext from ciphertext is called decryption. The accepted view among professional cryptographers (formalized in KIRKHOFF's law) is that the encryption algorithm should be published, whereas the key must be kept secret.

In the field of image cryptography, the focus has been put on steganography, and in particular on watermarking during the last years (seefor a review on watermarking). Watermarking, as opposed to steganography, has the additional requirement of robustness against possible image transformations. Watermarks are usually made invisible and should not be detectable.

In applications requiring transmission the image is first compressed, because it saves bandwidth.
Then the image is encrypted, as depicted in Figure 2.4.

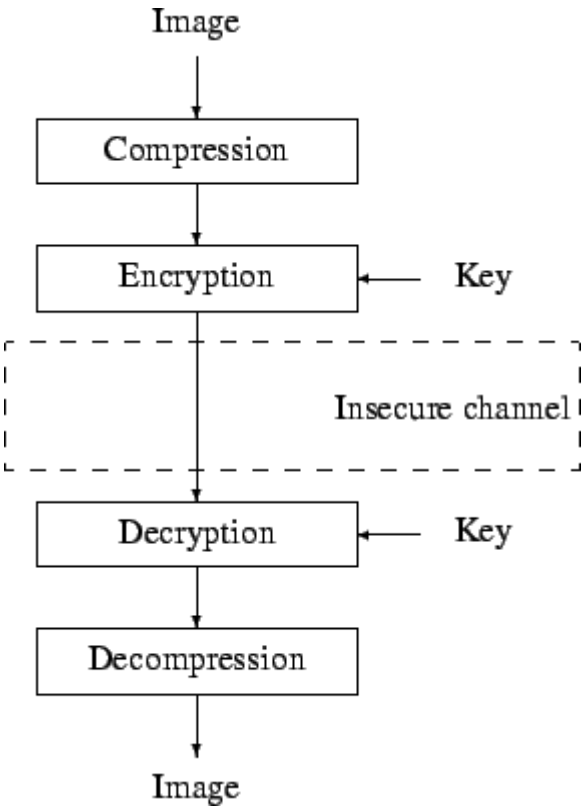


Figure 2.4: Encryption of an image.

The removal of redundancy enhances robustness as it squeezes out information that might be useful to a cryptanalyst. However it also introduces known patterns in the compressed bitstreams, like headers or synchronization stamps (called markers), that eases plaintext attacks on the signal. An alternative would be to compress after encryption, but it would not be as efficient in terms of bandwidth because encrypted information looks random and is therefore hard to compress. It is worth noting that, in schemes combining compression and encryption like the one shown in Figure 2.4

- there are two kinds of information: the image and the key.
- the subjective significance of information contained in the image is ignored. For example, there is no distinction between Most Significant Bits (MSBs) and Least Significant Bits (LSBs).

From Figure 2.4, it is clear that the receiver should decrypt the information before it can decompress the image. This approach has the main drawback that it is impossible to access the smallest part of information without knowledge of the key. For example, it would impossible to search through a general database of fully encrypted images. A way to address this issue is to use a technique called selective encryption; it is depicted in Figure 2.5.

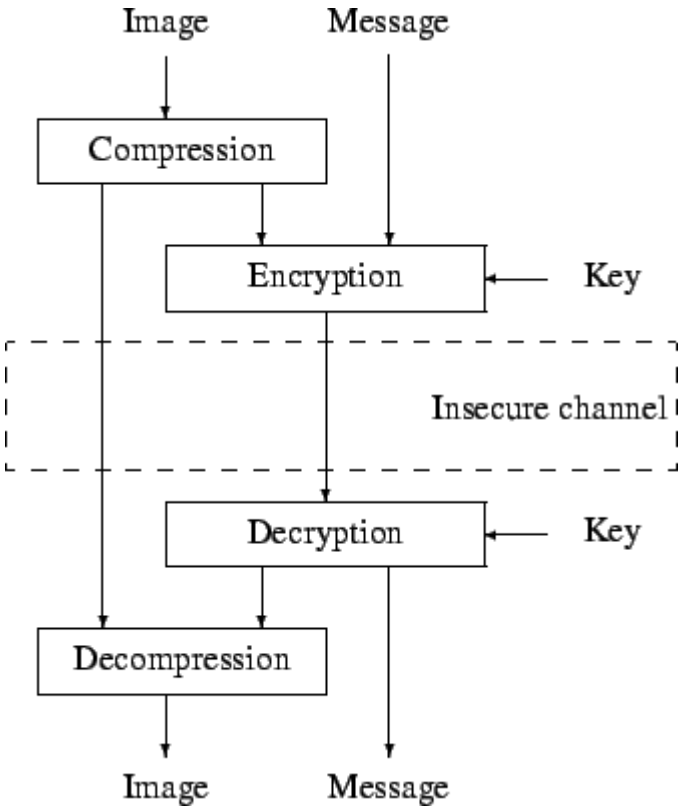


Figure 2.5: Selective encryption mechanism.

The general selective encryption mechanism works as follows. The image is first compressed (if needed). Afterwards the algorithm only encrypts part of the bitstream with a well-proven ciphering technique; incidentally a message (a watermark) can be added during this process. To guarantee a full compatibility with any decoder, the bitstream should only be altered at places where it does not compromise the compliance to the original format. This principle is sometimes referred to as format compliance. WEN et *al*. [] have recently described a general framework for format-compliant encryption. In their simulations, they focus on MPEG-4 video error resilient mode with data partitioning and discuss which fields can be encrypted. They also pointed out that the encryption of a variable length code (VLC) codeword may not result in another valid codeword.

With the decryption key, the receiver decrypts the bitstream, and decompresses the image. In principle, there should be no difference between a decoded image and an image that has been encrypted and decrypted. However there might be a slight though invisible difference if a watermark message has been inserted in the image.

When the decrypting key is unknown, the receiver will still be able to decompress the image, but this image will significantly differ from the original. This scenario is depicted in Figure

2.4.2 Analysis and Comparison of Image Encryption Algorithms

Shuqun Zhang and Mohammad A. Karim have proposed a new method to encrypt color images using existing optical encryption systems for gray-scale images. The color images are converted to their indexed image formats before they are encoded. In the encoding subsystem, image is encoded to stationary white noise with two random phase masks, one in the input plane and the other in the Fourier plane. At the decryption end, the color images are recovered by converting the decrypted indexed images back to their RGB (Red-GreenBlue) formats. The proposed single-channel color image encryption method is more compact and robust than the multichannels methods.

Visual Cryptography for Color Images :

Visual cryptography uses the characteristics of human vision to decrypt encrypted images. It needs neither cryptography knowledge nor complex computation. For security concerns, it also ensures that hackers cannot perceive any clues about a secret image from individual cover images. Young-Chang Hou have proposed three methods for visual cryptography. Gray-level visual cryptography method first transforms the gray-level image into a halftone image and then generates two transparencies of visual cryptography. Obviously, we indeed cannot detect any information about the secret image from the two sharing transparencies individually, but when stacking them together, the result clearly shows the secret image.

Method 1 uses four halftone images, cyan, magenta, yellow and black, to share the secret image. The codes of the four sharing images are fully disordered, and we cannot perceive any clue of the original secret image from any single sharing image. Method 2 reduces the inconvenience of Method 1 and requires only two sharing images to encrypt a secret image. However, after stacking the sharing images generated by Method 2, the range of color contrast will be 25% of that of the original image. Method 3 loses less image contrast, which is better than Method 2.

CHAPTER 3

PROBLEM FORMULATION

3.1 Introduction

The amount of satellite image has increased rapidly on the Internet, in public or local networks. Meteosat image security becomes increasingly important for many applications, e.g., confidential transmission, multispectral imaging for providing electronic images of clouds, land and sea surfaces, analysis of air masses to monitor the thermodynamic state in the lower part of the atmosphere and environment data collection and relay transmitted by automatic platforms (marine beacons, land and airborne ...). The unlawful, unofficial, and unauthorized access and illegal use of Meteosat imagery increases the importance of information security to keep the critical and confidential imagery and transmission process secure, dependable, trustworthy, and reliable. Cryptography is the most widely accepted information security technique employed to make the Meteosat image transmission processes reliable and secure from unauthorized access and illegal use. Cryptographic techniques can be divided into symmetric (with a secret key) and asymmetric encryption (with private and public keys).

3.2 Model Description

3.2.1 Transformation Technique

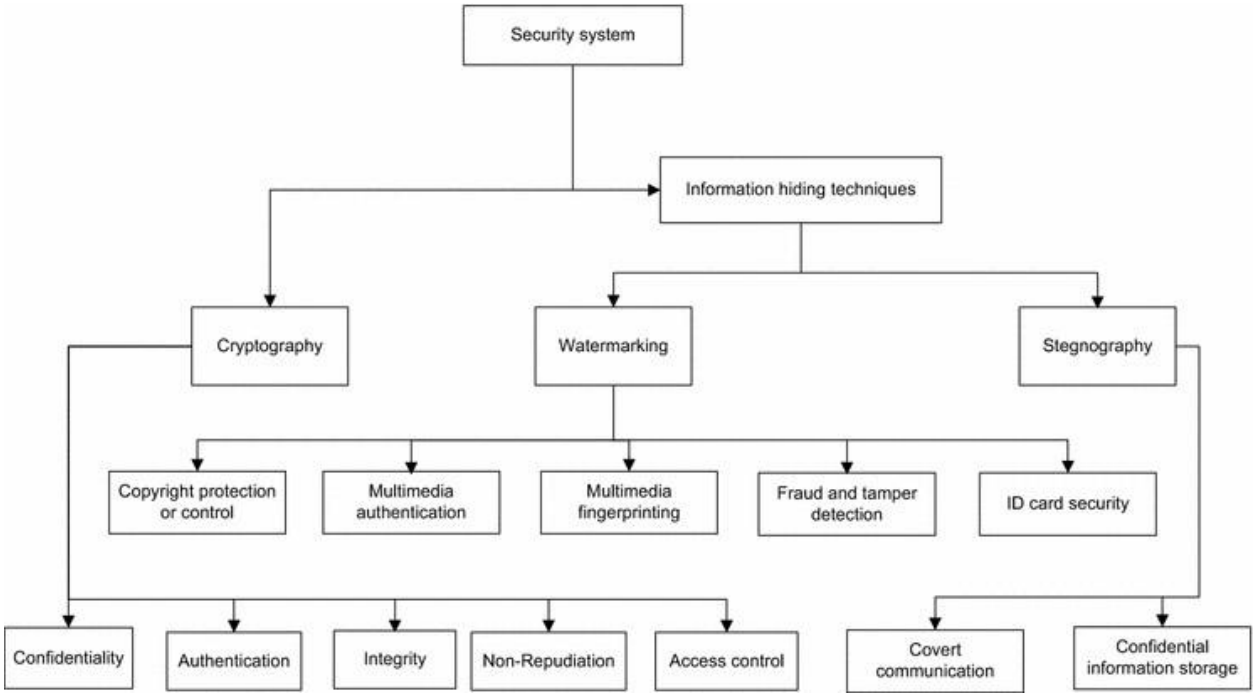


Figure 3.1 Transformation Techniques

3.2.2 Encryption Process

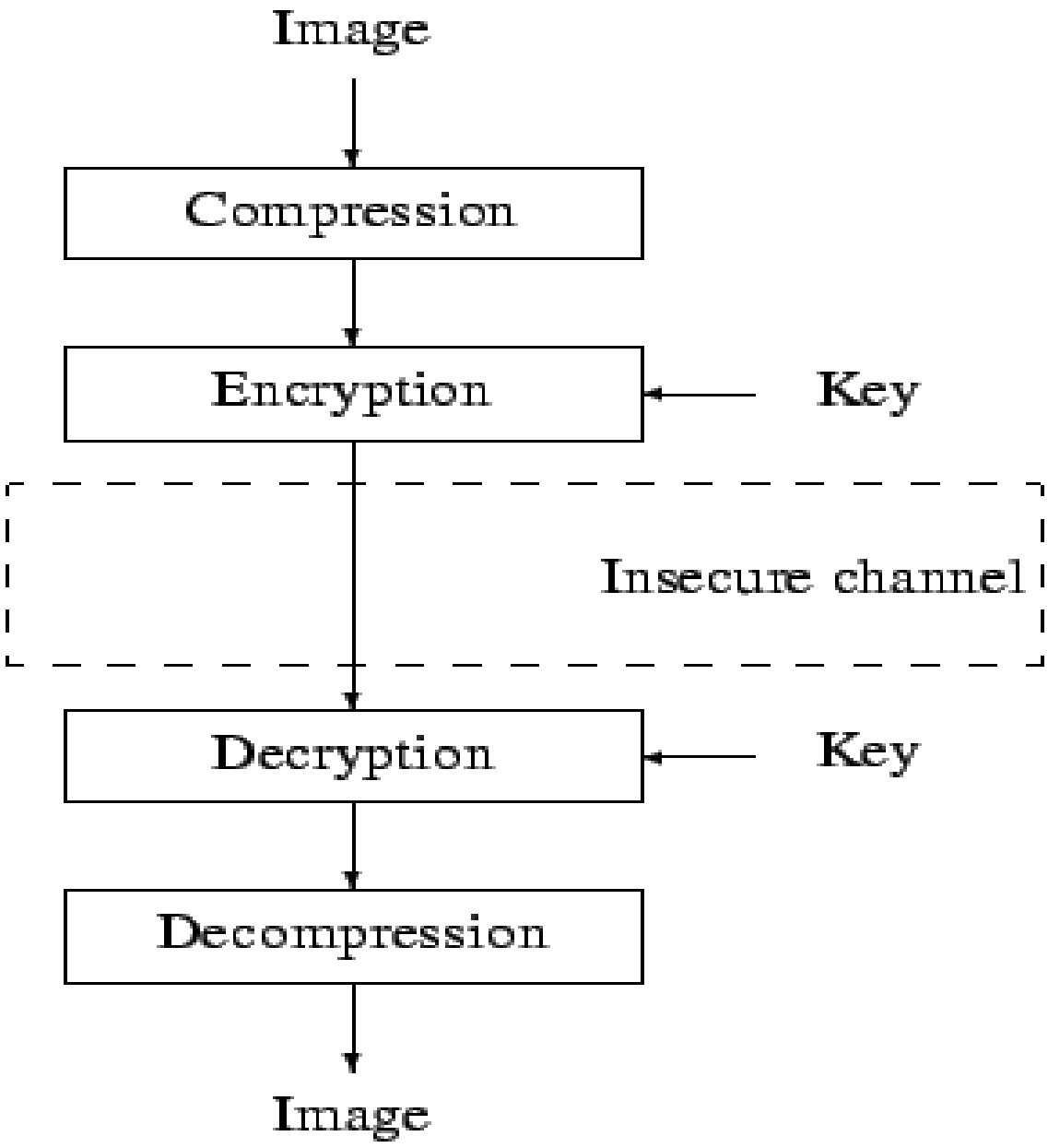


Figure 3.2: Encryption of an image.

3.3 Security Measures

3.3.1 Image Correlation

Digital image correlation (DIC) techniques have been increasing in popularity, especially in micro- and nano-scale mechanical testing applications due to its relative ease of implementation and use. Advances in computer technology and digital cameras have been the enabling technologies for this method and while white-light optics has been the predominant approach, DIC can be and has been extended to almost any imaging technology.

The concept of using cross-correlation to measure shifts in datasets has been known for a long time, and it has been applied to digital images since at least the early 1970s. The present-day applications are almost innumerable and include image analysis, image compression, velocimetry, and strain estimation. Much early work in DIC in the field of mechanics was led by researchers at the University of South Carolina in the early 1980S and has been optimized and improved in recent years Commonly, DIC relies on finding the maximum of the correlation array between pixel intensity array subsets on two or more corresponding images, which gives the integer translational shift between them. It is also possible to estimate shifts to a finer resolution than the resolution of the original images, which is often called "subpixel" registration because the measured shift is smaller than an integer pixel unit. For subpixel interpolation of the shift, there are other methods that do not simply maximize the correlation coefficient. An iterative approach can also be used to maximize the interpolated correlation coefficient by using nonlinear optimization techniques. The nonlinear optimization approach tends to be conceptually simpler, but as with most nonlinear optimization techniques, it is quite slow, and the problem can sometimes be reduced to a much faster and more stable linear optimization in phase space.

3.3.2 Image histogram

An image histogram is a type of histogram that acts as a graphical representation of the tonal distribution in a digital image. It plots the number of pixels for each tonal value. By looking at the histogram for a specific image a viewer will be able to judge the entire tonal distribution at a glance.

Image histograms are present on many modern digital cameras. Photographers can use them as an aid to show the distribution of tones captured, and whether image detail has been lost

to blown-out highlights or blacked-out shadows. This is less useful when using a raw image format, as the dynamic range of the displayed image may only be an approximation to that in the raw file.

The horizontal axis of the graph represents the tonal variations, while the vertical axis represents the number of pixels in that particular tone. The left side of the horizontal axis represents the black and dark areas, the middle represents medium grey and the right hand side represents light and pure white areas. The vertical axis represents the size of the area that is captured in each one of these zones. Thus, the histogram for a very dark image will have the majority of its data points on the left side and center of the graph. Conversely, the histogram for a very bright image with few dark areas and/or shadows will have most of its data points on the right side and center of the graph.

3.3.3 Image Entropy

In information theory, systems are modeled by a transmitter, channel, and receiver. The transmitter produces messages that are sent through the channel. The channel modifies the message in some way. The receiver attempts to infer which message was sent. In this context, entropy (more specifically, Shannon entropy) is the expected value (average) of the information contained in each message. 'Messages' can be modeled by any flow of information.

In a more technical sense, there are reasons (explained below) to define information as the negative of the logarithm of the probability distribution. The probability distribution of the events, coupled with the information amount of every event, forms a random variable whose expected value is the average amount of information, or entropy, generated by this distribution. Units of entropy are the shannon, nat, or hartley, depending on the base of the logarithm used to define it, though the shannon is commonly referred to as a bit.

The logarithm of the probability distribution is useful as a measure of entropy because it is additive for independent sources. For instance, the entropy of a coin toss is 1 shannon, whereas of m tosses it is m shannons. Generally, you need $\log_2(n)$ bits to represent a variable that can take one of n values if n is a power of 2. If these values are equally probable, the entropy (in shannons) is equal to the number of bits. Equality between number of bits and shannons holds only while all outcomes are equally probable. If one of the events is more probable than others, observation of that event is less informative. Conversely, rarer events provide more information when observed. Since observation of less probable events occurs

more rarely, the net effect is that the entropy (thought of as average information) received from non-uniformly distributed data is less than $\log_2(n)$. Entropy is zero when one outcome is certain. Shannon entropy quantifies all these considerations exactly when a probability distribution of the source is known. The meaning of the events observed (the meaning of messages) does not matter in the definition of entropy. Entropy only takes into account the probability of observing a specific event, so the information it encapsulates is information about the underlying probability distribution, not the meaning of the events themselves.

3.3.4 Image Similarity

Metrics are probably the most critical element of a registration problem. The metric defines what the goal of the process is, they measure how well the Target object is matched by the Reference object after the transform has been applied to it. The Metric should be selected in function of the types of objects to be registered and the expected kind of misalignment. Some metrics has a rather large capture region, which means that the optimizer will be able to find his way to a maximum even if the misalignment is high. Typically large capture regions are associated with low precision for the maximum. Other metrics can provide high precision for the final registration, but usually require to be initialized quite close to the optimal value.

Unfortunately there are no clear rules about how to select a metric, other that trying some of them in different conditions. In some cases could be and advantage to use a particular metric to get an initial approximation of the transformation, and then switch to another more sensitive metric to achieve better precision in the final result.

Metrics are depend on the objects they compare. The toolkit currently offers *Image*
To Image and PointSet *to* Image metrics as follows:

- **Mean Squares** Sum of squared differences between intensity values. It requires the two objects to have intensity values in the same range.
- **Normalized Correlation** Correlation between intensity values divided by the square rooted

  autocorrelation of both target and reference objects: $\frac{\sum_i^n a_i * b_i}{\sum_i^n a_i^2 \sum_i^n b_i^2}$. This metric allows to register objects whose intensity values are related by a linear transformation.

- **Pattern Intensity** Squared differences between intensity values transformed by a function of type $\frac{1}{1+x}$ and summed them up. This metric has the advantage of increase simultaneously when more samples are available and when intensity values are close.
- **Mutual Information** In probability theory and information theory, the **mutual information** (**MI**) of two random variables is a measure of the mutual dependence between the two variables. More specifically, it quantifies the "amount of information" (in units such as bits) obtained about one random variable, through the other random variable. The concept of mutual information is intricately linked to that ofentropy of a random variable, a fundamental notion in information theory, that defines the "amount of information" held in a random variable.

  Not limited to real-valued random variables like the correlation coefficient, MI is more general and determines how similar the joint distribution $p(X,Y)$ is to the products of factored marginal distribution $p(X)p(Y)$. MI is the expected value of the pointwise mutual information (PMI). The most common unit of measurement of mutual information is the bit.

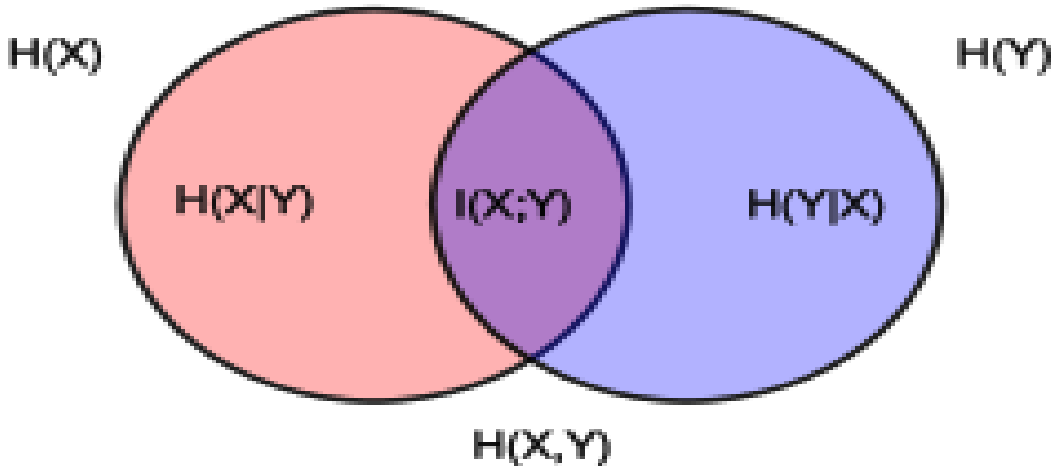


Figure 3.3 Mutual Information

3.3.5 Decryption Process:

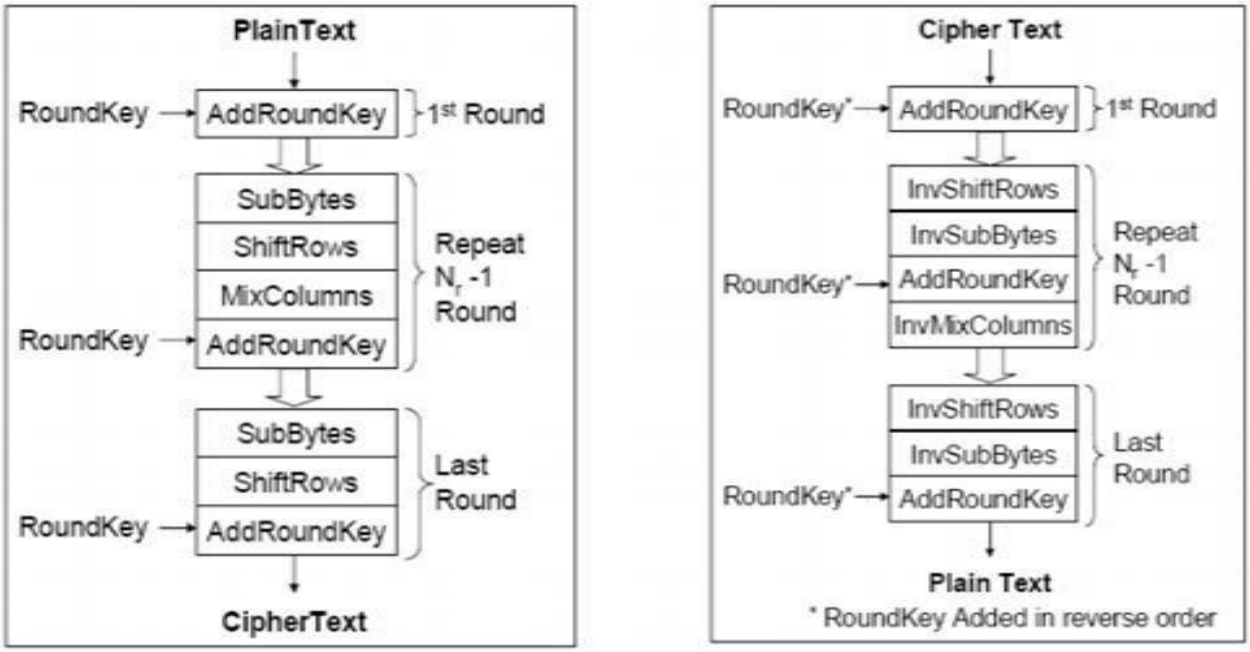


Figure 3.4 : Decryption Process

CHAPTER 4

PROPOSED WORK

4.1 Comparison of current algorithms

A brief comparison of image encryption schemes is given . Also detailed properties of each method are introduced following.

Image Encryption using Digital Signatures algorithm encrypts the image and embeds the digital signature into the image prior to transmission. This encryption technique provides three layers of security. In the first step, an error control code is used which is determined in real-time, based on the size of the input image. Without the knowledge of the specific error control code, it is very difficult to obtain the original image. The dimension of the image also changes due to the added redundancy. This poses an additional difficulty to decrypt the image. Also, the digital signature is added to the encoded image in a specific manner. At the receiver end, the digital signature can be used to verify the authenticity of the transmitted image. The advantage of the scheme is the authenticity verification. Increment in the size of the image due to added redundancy is the disadvantage of the algorithm. Also it does not have any compression scheme.

The algorithm which uses SCAN language has lossless image compression and encryption abilities. The distinct advantage of simultaneous lossless compression and strong encryption makes the methodology very useful in applications such as medical imaging, multimedia applications, and military applications. The drawback of the methodology is that compression-encryption takes longer time.

Mirror-like image encryption algorithm and chaotic image encryption algorithm are similar in nature. Both algorithms use binary sequence generated from the chaotic system to rearrange an image pixels.

These algorithms do not have any compression scheme and authenticity verification. However they do lossless image encryption-decryption which makes images to be in a chaotic state very quickly.

Major advantage of the algorithm which is proposed by Chin-Chen Chang, Min-Shian Hwang, and TungShou Chen has a simple hardware structure. Required bit rate of VQ is

Since VQ compresses the original image into a set of indices in the codebook, we can save a lot of storage space and channel bandwidth. The other advantage is that VQ has a simple hardware structure for providing a fast decoding procedure.

Color Image Encryption Using Double Random Phase Encoding technique introduces color information to optical encryption. An RGB color image is converted to an indexed image before it is encrypted using a typical optical security systems. At the decryption end, the recovered indexed image is converted back to the RGB image. Since only one channel is needed to encrypt color images, it reduces the complexity and increases the reliability of the corresponding optical color image encryption systems. The most notable feature of the visual cryptography for color images approach is that it can recover a secret image without any computation. It exploits the human visual system to read the secret message from some overlapping shares, thus overcoming the disadvantage of complex computation required in the traditional cryptography. Also, the contrast of the stacked image is somewhat downgraded, but the content of the image can still be easily identified.

4.2 Adding compression to MIE and VC algorithms

We have compared the MIE and VC algorithms. Results are given also enhancements which are shown in Fig4.1 and Fig4. 2 have been done on the MIE and VC algorithms by adding compression ability to them. In the new enhanced scheme encrypted images are compressed by either loss or lossless compression algorithms before transmission to the destination.

We have modified MIE algorithm so that it can reduce the disk storage space and network bandwidth. Before transmission encrypted image is compressed with the compression algorithm. At the receiver end compressed image is decompressed and then decrypted. Also compression is added for visual cryptography. Visual cryptography produces 2 sharing images for gray-level images, 4 sharing images for method1, 2 sharing images for method2 and method3. Compression is more important issue for visual cryptography because it produces 2 or more sharing images which are twice in size of dimensions of the original image.

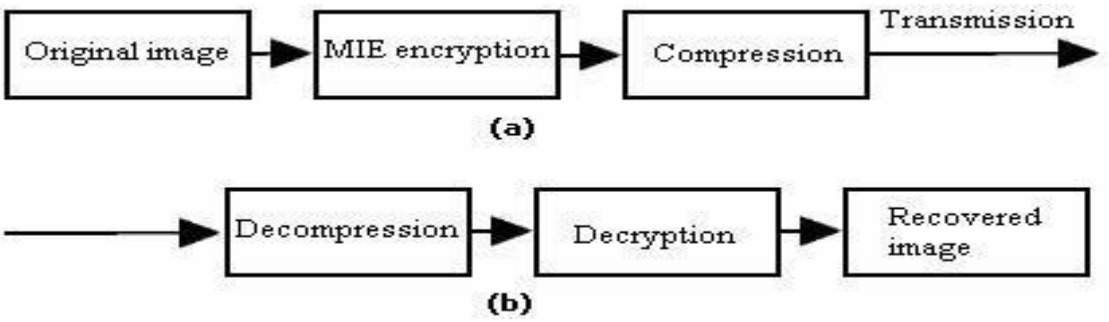


Figure 4.1. New MIE (a) at sender end (b) at receiver end

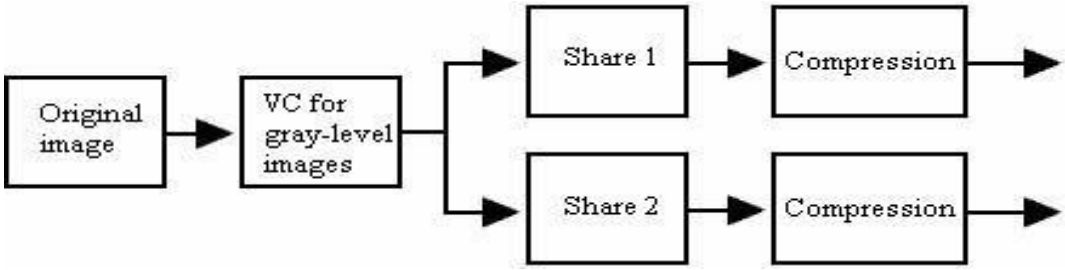


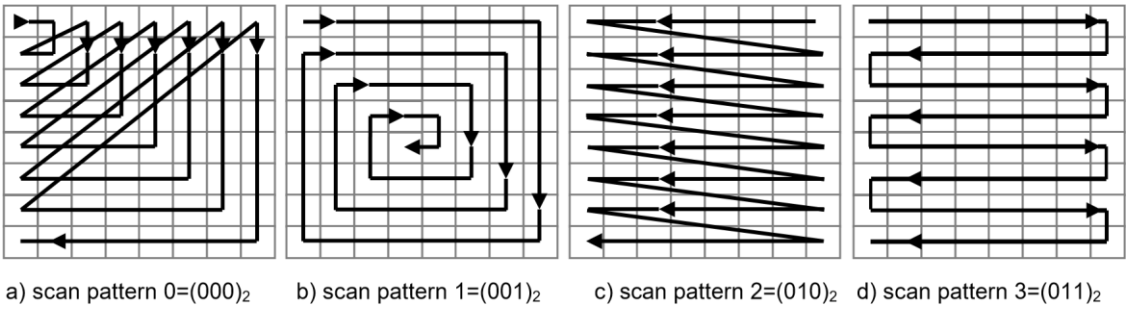
Figure 4.2. New VC for gray-level images encryption

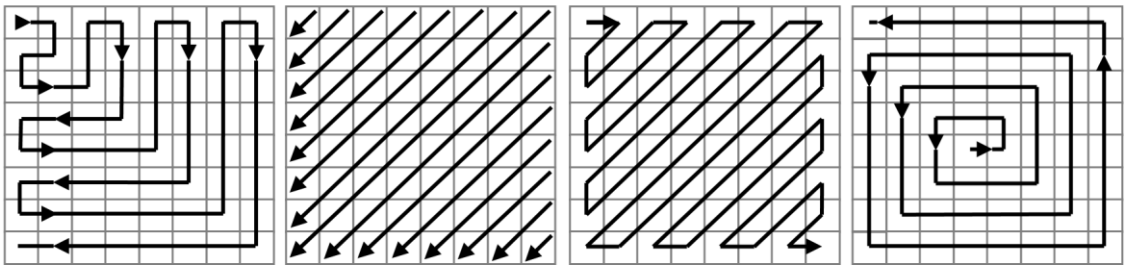
JPEG (Joint Photographic Experts Group) is a standard compression algorithm used to reduce memory requirement for the storage of digital images. The JPEG standard allows to specify the desired quality of the encoded image by varying a quality factor between 0 (lowest quality) and 100 (best quality).

PNG is an extensible file format for the lossless, portable, well-compressed storage of raster images. While the jpeg compression has losses in the compressed image, in PNG compression there is neither a change of colors nor a reduction of color depth.

Mean square error (MSE) is the cumulative squared error between original and recovered image. Lower value of MSE means lesser error.

4.3 Lossless Image Compression and Encryption Using SCAN





e) scan pattern 4=$(100)_2$ f) scan pattern 5=$(101)_2$ g) scan pattern 6=$(110)_2$
h) scan pattern 7=$(111)_2$

Figure 4.3. Different Scan Patterns Indexed from 0=$(000)_2$ to 7=$(111)_2$

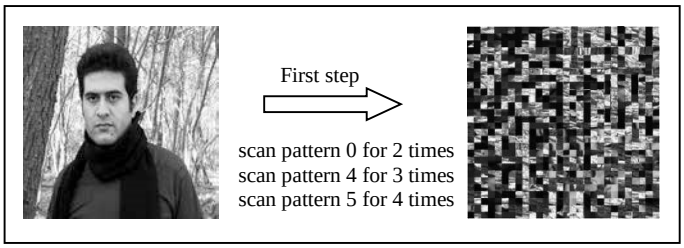
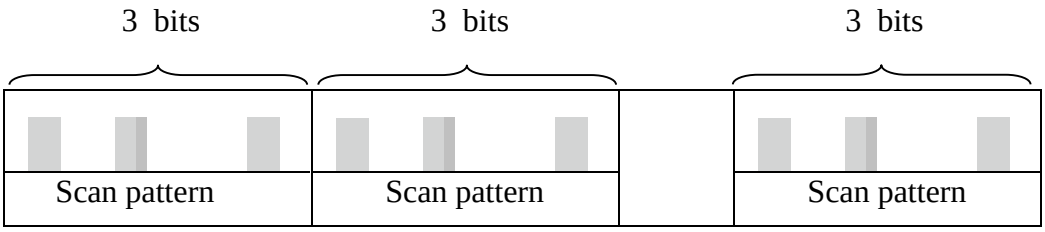


Figure 4.4. Whole 8*8 Blocks of Original Image Shuffled by 3 Scan Patterns for Desired Times

Formation of key (48 bits): 8 different Scan patterns need 3 bits to be indexed. Each scan pattern can be run from 0 to 7 times, hence it needs 3 bits. Each scan pattern conjunction with number of its running needs 3+3=6 bits. Therefore 8(number of scan patterns) * 6 = 48 bits is enough for addressing desired repositioning.



$(000)_2$ or $(00\ 1)_2$ or … or$111)_2$ $(-000)_2$ or $(00\ 1)_2$ or … or$111)_2$ $(-000)_2$ or $(\ 00\ 1)_2$ or … or $111)_2$

Allowable Number of run Allowable Number of run Allowable Number of run

$(000)_2$ t0 $(111)_2$ $(000)_2$ t0 $(111)_2$ $(000)_2$ t0 $(111)_2$

CHAPTER 5

SYSTEM DESIGN AND INITIAL IMPLEMENTATION

5.1 Tools:

MATLAB, short for MATrix LABoratory is a programming package specifically designed for quick and easy scientific calculations and I/O. It has literally hundreds of built-in functions for a wide variety of computations and many toolboxes designed for specific research disciplines, including statistics, optimization, solution of partial differential equations, data analysis.

5.1.1 Introduction to MATLAB graphics

MATLAB has a large number of functions associated with graphical output. If you'd like to explore the possibilities use **help plot** or **help plot3** for 3-dimensional plots, or run the MATLAB demo (by typing **demo**) and look at the information on visualization and graphics. We start with basic plotting routines and look at some fancy graphics to get a taste of MATLAB's abilities.

Suppose that the maximum and minimum temperature (in degrees Celsius) recorded from 12 to 18 December are 23, 27, 21, 28, 24, 25, 26, and 11, 10, 15, 15, 14, 15, 12 respectively.

1)Set up the vector 'date' to have elements from 12 to 18, and the vectors 'maxtemp' and 'mintemp' to contain the temperature data given above. It does not matter whether the vectors are row or column vectors.

All the vectors are of the same length so it is possible to plot one against the other.

2)Create a plot of the maximum temperatures by typing **plot(date, maxtemp)**

The x-axis variable is listed first. This graph will be created in a window called Figure No. 1. If this window is not visible select Figure No. 1 under Window on the menu. Your graph should look like the one below.

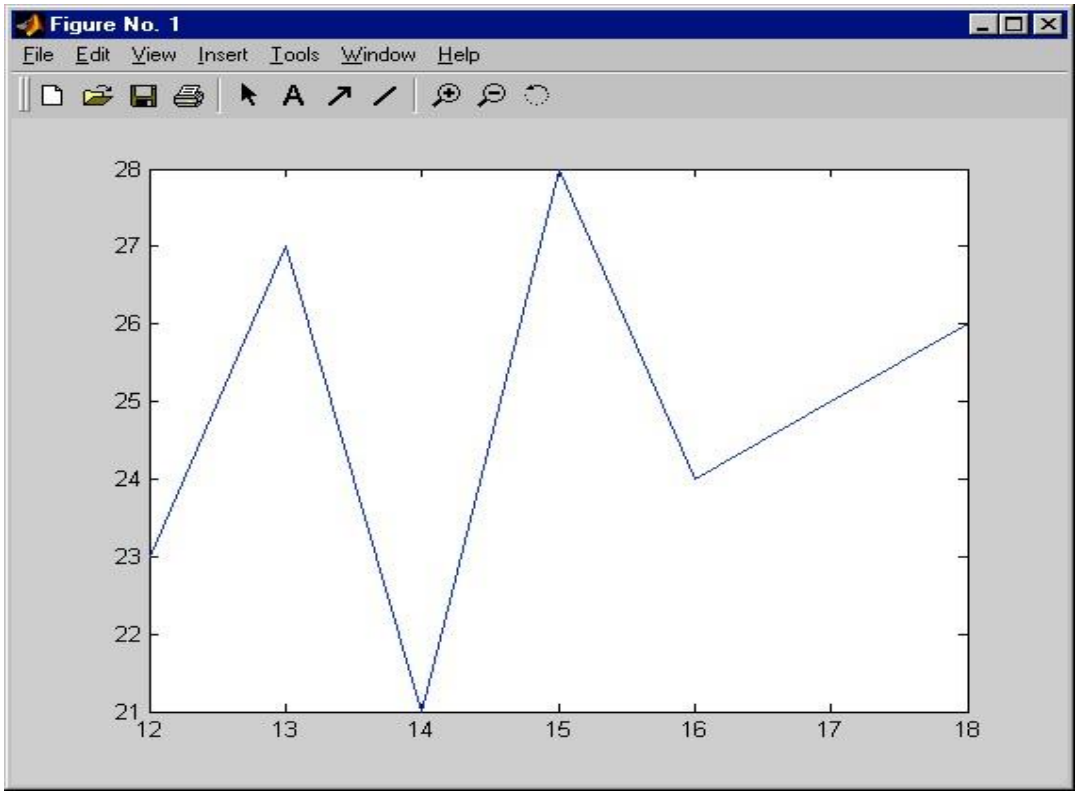


Figure 5.1 Ploting graph on MATLAB

3) Now type **plot(date,mintemp)** . The previous plot disappears and the new plot is displayed. We can also plot both graphs in one figure:

4) Type **plot(date,maxtemp);** followed by **hold on;** Then type **plot(date, mintemp);** followed by **hold off; hold on** tells MATLAB to keep the old plot and add the new graph to it. **hold off** turns the hold-feature off again. We could also have used **plot(date, maxtemp, date, mintemp);** which tells MATLAB to plot maxtemp against date and then mintemp against date in the same graph. The colours of the graphs are then different.

The plot could be made to look a lot better.

5) Select Insert from the figure window and then Title. A text box will appear in the figure window. Type '12-18 December'. Now add the label 'date' to the x-axis and 'temp (Celsius)' to the y-axis using the appropriate selections under Insert.

Note: in older versions of MATLAB a title can be created using the MATLAB command **title** and labels can be added using the **xlabel** and **ylabel** commands.

The set of axes chosen is just large enough to contain all the data points, but this doesn't necessarily produce the most pleasing display.

6) Select Edit and then Axes properties. Un-select Auto for X (click in the Auto box to remove the tick). Then change the limits of 12 and 18 for the x-axis to 11.5 and 18.5 respectively.

Now change the limits for the y-axis to 9 and 30. If the Immediately apply box at the bottom of the Axes Properties window is not ticked you must click OK to see the changes.

This plot is better but some might say that discrete data points should not be joined with lines because, for example, 'maximum temperature on the twelfth-and-a-half day in December' is not a meaningful statement.

7) Click on the graph corresponding to the maximum temperature. Then select Current Object Properties under Edit in the figure window. Select No line (none) under Line Style. Under Marker Properties, select a marker of your choice, eg. Six-pointed star. Choose a nice colour.

8) Repeat for the graph corresponding to the minimum temperature with a different colour and marker.If you want to change two or more graphs at the same time (and give them the same properties), you click on the first graph, hold down the shift key while clicking on the other graphs and then follow the above steps.

9) To add a legend, select Insert then Legend. You can change the description of the graphs in the legend by double clicking on the text in the figure window. Change the descriptions to 'maxima' for the maximum temperature graph and 'minima' for the minimum temperature graph. Finally we add some text inside the plot.

10) Select Insert, then Text (or click on the A button on the menu). Move your mouse to the plot window. Position the cursor near the highest point in the graph and click. Type 'highest' in the field. The final graph is shown below

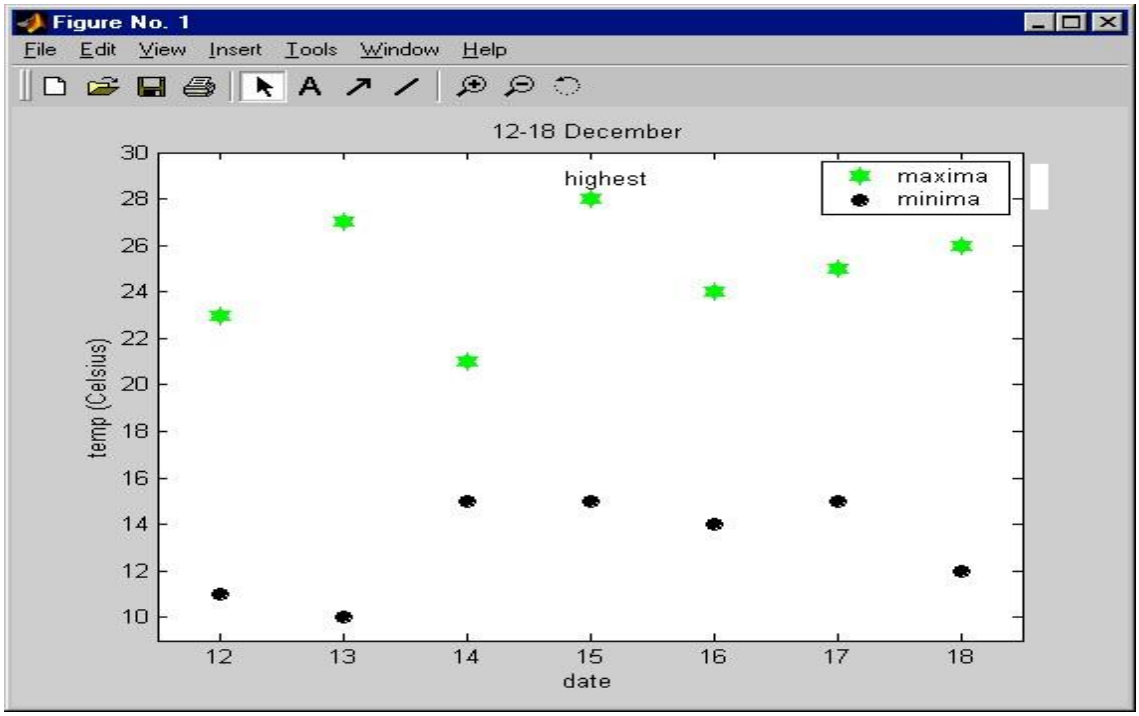


Figure 5.2 Plotting graph on MATLAB

5.2 Symmetric Encryption:

One type of encryption is symmetric encryption. It also called as single key cryptography. It uses a single key also known as private key that is used for both encryption and decryption sometimes also called as secret key. The sender can encrypt data with private key and send it to receiver who can decrypt data with that key. Symmetric Encryption is very common in database applications. Symmetric Encryption is very fast as compared to asymmetric encryption. Symmetric encryption covers a number of algorithms such as Blowfish, AES, DES and 3DES which are still in use.

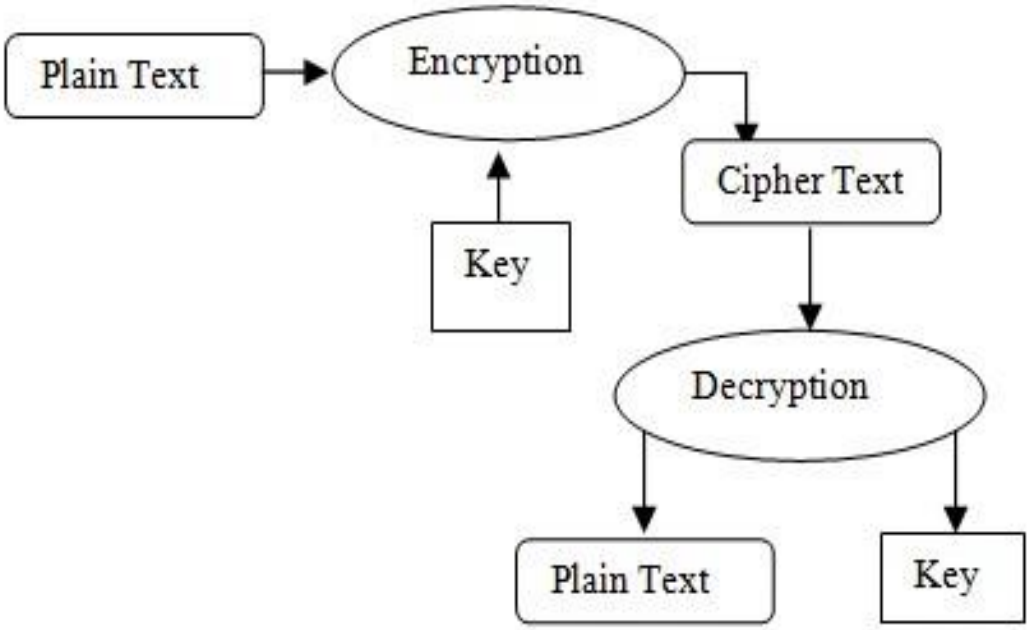


Fig. 5.3 Symmetric Encryption

Symmetric key algorithms are used primarily for the bulk encryption of data or data streams. These algorithms are designed to be very fast and have a large number of possible keys. The best symmetric key algorithms offer excellent secrecy; once data is encrypted with a given key, there is no fast way to decrypt the data without possessing the same key.

Symmetric key algorithms can be divided into two categories: block and stream. Block algorithms encrypt data a block (many bytes) at a time, whilestream algorithms encrypt byte by byte (or even bit by bit).

5.2.1 Cryptographic Strength of Symmetric Algorithms

Different encryption algorithms are not equal. Some systems are not very good at protecting data, allowing encrypted information to be decrypted without knowledge of the requisite key. Others are quite resistant to even the most determined attack. The ability of a cryptographic system to protect information from attack is called its strength. Strength depends on many factors, including:

- The secrecy of the key.
- The difficulty of guessing the key or trying out all possible keys (a key search). Longer keys are generally more difficult to guess or find.
- The difficulty of inverting the encryption algorithm without knowing the encryption key (breaking the encryption algorithm).
- The existence (or lack) of back doors, or additional ways by which an encrypted file can be decrypted more easily without knowing the key.
- The ability to decrypt an entire encrypted message if you know how a portion of it decrypts (called a known plaintext attack).
- The properties of the plaintext and knowledge of those properties by an attacker. For example, a cryptographic system may be vulnerable to attack if all messages encrypted with it begin or end with a known piece of plaintext. These kinds of regularities were used by the Allies to crack the German Enigma cipher during World War II.

In general, cryptographic strength is not proven; it is only disproven. When a new encryption algorithm is proposed, the author of the algorithm almost always believes that the algorithm offers "perfect" security?that is, the author believes there is no way to decrypt an encrypted message without possession of the corresponding key. After all, if the algorithm contained a known flaw, then the author would not propose the algorithm in the first place (or at least would not propose it in good conscience).

As part of proving the strength of an algorithm, a mathematician can show that the algorithm is resistant to specific kinds of attacks that have been previously shown to compromise other algorithms. Unfortunately, even an algorithm that is resistant to every known attack is not necessarily secure, because new attacks are constantly being developed.

From time to time, some individuals or corporations claim that they have invented new symmetric encryption algorithms that are dramatically more secure than existing algorithms. Generally, these algorithms should be avoided. As there are no known attack methods against the encryption algorithms that are in wide use today, there is no reason to use new, unproven encryption algorithms that might have flaws lurking in them.

5.2.2 Key Length with Symmetric Key Algorithms

Among those who are not entirely familiar with the mathematics of cryptography, key length is a topic of continuing confusion. As we have seen, short keys can significantly compromise the security of encrypted messages because an attacker can merely decrypt the message with every possible key to decipher the message's content. But while short keys provide comparatively little security, extremely long keys do not necessarily provide significantly more practical security than keys of moderate length. That is, while keys of 40 or 56 bits are not terribly secure, a key of 256 bits does not offer significantly more real security than a key of 168 bits, or even a key of 128 bits.

To understand this apparent contradiction, it is important to understand what is really meant by the words key length, and how a brute force attack actually works.

Inside a computer, a cryptographic key is represented as a string of binary digits. Each binary digit can be a 0 or a 1. Thus, if a key is 1 bit in length, there are two possible keys: 0 and 1. If a key is 2 bits in length, there are four possible keys: 00, 01, 10, and 11. If a key is 3 bits in length, there are eight possible keys: 000, 001, 010, 011, 100, 101, 110, and 111. In general, each added key bit doubles the number of keys. The mathematical equation that relates the number of possible keys to the number of bits is:

$$\text{number of keys} = 2^{(\text{number of bits})}$$

If you are attempting to decrypt a message and do not have a copy of the key, the simplest way to decrypt the message is to do a brute force attack. These attacks are also called key search attacks, because they involve trying every possible key to see if a specific key decrypts the message. If the key is selected at random, then on average, an attacker will need to try half of all the possible keys before finding the actual decryption key.

Fortunately, for those of us who depend upon symmetric encryption algorithms, it is a fairly simple matter to use longer keys. Each time a bit is added, the difficulty for an attacker attempting a brute force attack doubles.

The first widely used encryption algorithm, the DES, used a key that was 56 bits long. At the time that the DES was adopted, many academics said that 56 bits was not sufficient: they argued for a key that was twice as long. But it has been conjectured that the U.S. National Security Agency did not want a cipher with a longer key length widely deployed, most likely because such a secure cipher would significantly complicate its job of international surveillance. To further reduce the impact that the DES would have on its ability to collect international intelligence, U.S. corporations were forbidden from exporting products that implemented the DES algorithm. The NSA operates a worldwide intelligence surveillance network. This network relies, to a large extent, on the fact that the majority of the information transmitted electronically is transmitted without encryption. The network is also used for obtaining information about the number of messages exchanged between various destinations, a technique called traffic analysis. Although it is widely assumed that the NSA has sufficient computer power to forcibly decrypt a few encrypted messages, not even the NSA has the computer power to routinely decrypt all of the world's electronic communications.

In the early 1990s, a growing number of U.S. software publishers demanded the ability to export software that offered at least a modicum of security. As part of a compromise, a deal was brokered between the U.S. Department of Commerce, the National Security Agency, and the Software Publisher's Association. Under the terms of that agreement, U.S. companies were allowed to export mass-market software that incorporated encryption, provided that the products used a particular encryption algorithm and the length of the key was limited to 40 bits. At the same time, some U.S. banks started using an algorithm called Triple-DES (basically, a threefold application of the DES algorithm) to encryp some financial transactions. It has a key size of 168 bits. Triple-DES is described in the following section.

In October 2000, the National Institute of Standards and Technology (NIST) approved the Rijndael encryption algorithm as the new U.S. Advanced Encryption Standard. Rijndael can be used with keys of 128, 192, or 256 bits. The algorithm's extremely fast speed, combined with its status as the government-chosen standard, means that it will likely be preferable to the DES, Triple-DES, and other algorithms in the future.

So how many bits is enough? That depends on how fast the attacker can try different keys and how long you wish to keep your information secure. As Table 7-1 shows, if an attacker can try only 10 keys per second, then a 40-bit key will protect a message for more than 3,484 years. Of course, today's computers can try many thousands of keys per second?and with special-purpose hardware and software, they can try hundreds of thousands. Key search speed can be further improved by running the same program on hundreds or thousands of computers at a time. Thus, it's possible to search a million keys per second or more using today's technology. If you have the ability to search a million keys per second, you can try all 40-bit keys in only 13 days.

If a key that is 40 bits long is clearly not sufficient to keep information secure, how many bits are necessary? In April 1993, the Clinton Administration introduced the Clipper encryption chip as part of its Escrowed Encryption Initiative (EEI). This chip used a key that was 80 bits long. As Table 7-1 shows, an 80-bit key is more than adequate for many applications. If you could search a billion keys per second, trying all 80-bit keys would still require 38 million years! Clipper was widely criticized not because of the key length, but because the Clipper encryption algorithm was kept secret by the National Security Agency, and because each Clipper chip came with a "back door" that allowed information encrypted by each Clipper chip to be decrypted by the U.S. government in support of law enforcement and intelligence needs.

5.2.3 Types of Symmetric Encryption Algorithms

Algorithms for encrypting computer data come in two main varieties: symmetric and asymmetric. Each encryption type has inherent strengths and weaknesses. Symmetric algorithms convert plain-text data into an unreadable ciphertext using a single key or password; they decrypt the ciphertext using the same key. These algorithms are relatively simple and quick, but if third parties intercept the key they can decrypt the messages. The need for trustworthy e-commerce and computer-file security has led researchers to develop several types of encryption algorithms.

DES and Triple DES

The Triple Data Encryption Standard, or Triple DES algorithm, evolved from the original DES algorithm introduced as a standard in 1976 (Reference 2, page 3). DES uses 56 bits of a 64-bit key to encrypt messages in fixed-sized blocks of data. Though considered secure in the 1970s, advances in computing speed led to sophisticated attacks breaking DES encryption in the late 1990s (Reference 2, page 6). Because researchers found DES vulnerable, software developers instead use a newer standard, Triple DES.

The new standard increases the strength of the algorithm by using two or three 64-bit keys and performing encryption three times on each message. The results of each pass are used as the source for the next one.

RC2

Ron Rivest developed the RC2 algorithm in the late 1980s as a replacement for DES. RC2 encrypts data in 64-bit blocks and has a variable key size of 8 to 128 bits in 8-bit increments. Lotus Development requested Rivest's help in creating RC2 for the company's Lotus Notes software. Because a large part of an encryption algorithm's strength lies in the length of its keys, researchers now consider RC2 to be too easily compromised (Reference 3).

Blowfish and Twofish

Security researcher Bruce Schneier developed the symmetric algorithm "Blowfish" in the early 1990s (Reference 3). As with RC2, Blowfish breaks messages up into equal-sized 64-bit blocks and encrypts the blocks. Its key sizes range from 32 to 448 bits. Schneier released Blowfish as a public-domain algorithm, freely available to anyone wanting to encrypt data. Seeking to improve upon Blowfish, he later developed Twofish, which uses 128-bit blocks and keys up to 256 bits long. Twofish is one of the fastest fixed-block algorithms currently available, and though it has theoretical vulnerabilities, no one has yet broken it.

Serpent

Cambridge researchers Ross Anderson, Eli Biham, and Lars Knudsen developed the Serpent algorithm in 2000 (Reference 4). The researchers believed other algorithms had theoretical flaws that rendered their encryption vulnerable to shortcut attacks. They sought to develop an encryption algorithm that was as free from these flaws as possible. Serpent was the result of their efforts; it uses a 128-bit block and 256-bit keys (Reference 5). As with Blowfish and Twofish, the Serpent algorithm is in the public domain. Researchers have given Serpent very high scores for "safety factor," or trustworthiness against attack.

5.3 Asymmetric Encryption:

It is also called as public key cryptography. It uses two keys: public key and a private key, public and private key have unique characteristics in asymmetric encryption in which we can encrypt with public key and there is matching private key which is used for decryption. Typically there is a public key which the sender use to encrypt data, and data is send in encrypted format to the receiver who uses the private key to decrypt the data.

Thus asymmetric encryption uses a pair of keys, public and private to encrypt the information.

We can encrypt with public key and decrypt with private key (vice versa). Asymmetric encryption works very well when there is two different end points. Examples of asymmetric encryption are web browsers, VPN, secure FTP.

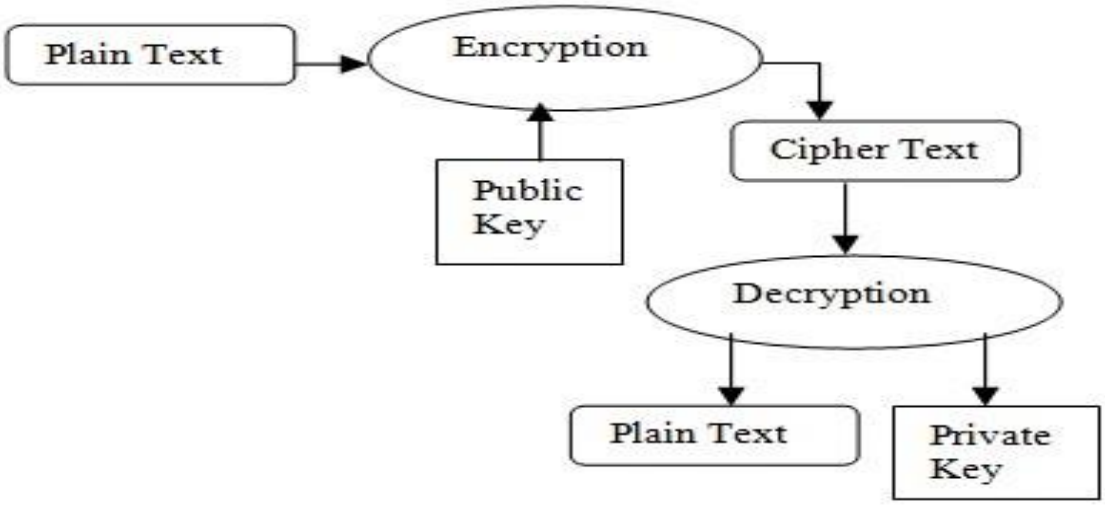


Fig. 5.4 Asymmetric Encryption

Unlike symmetric key cryptography, we do not find historical use of public-key cryptography. It is a relatively new concept.

Symmetric cryptography was well suited for organizations such as governments, military, and big financial corporations were involved in the classified communication.

With the spread of more unsecure computer networks in last few decades, a genuine need was felt to use cryptography at larger scale. The symmetric key was found to be non-practical due to challenges it faced for key management. This gave rise to the public key cryptosystems.

The process of encryption and decryption is depicted in the following illustration –

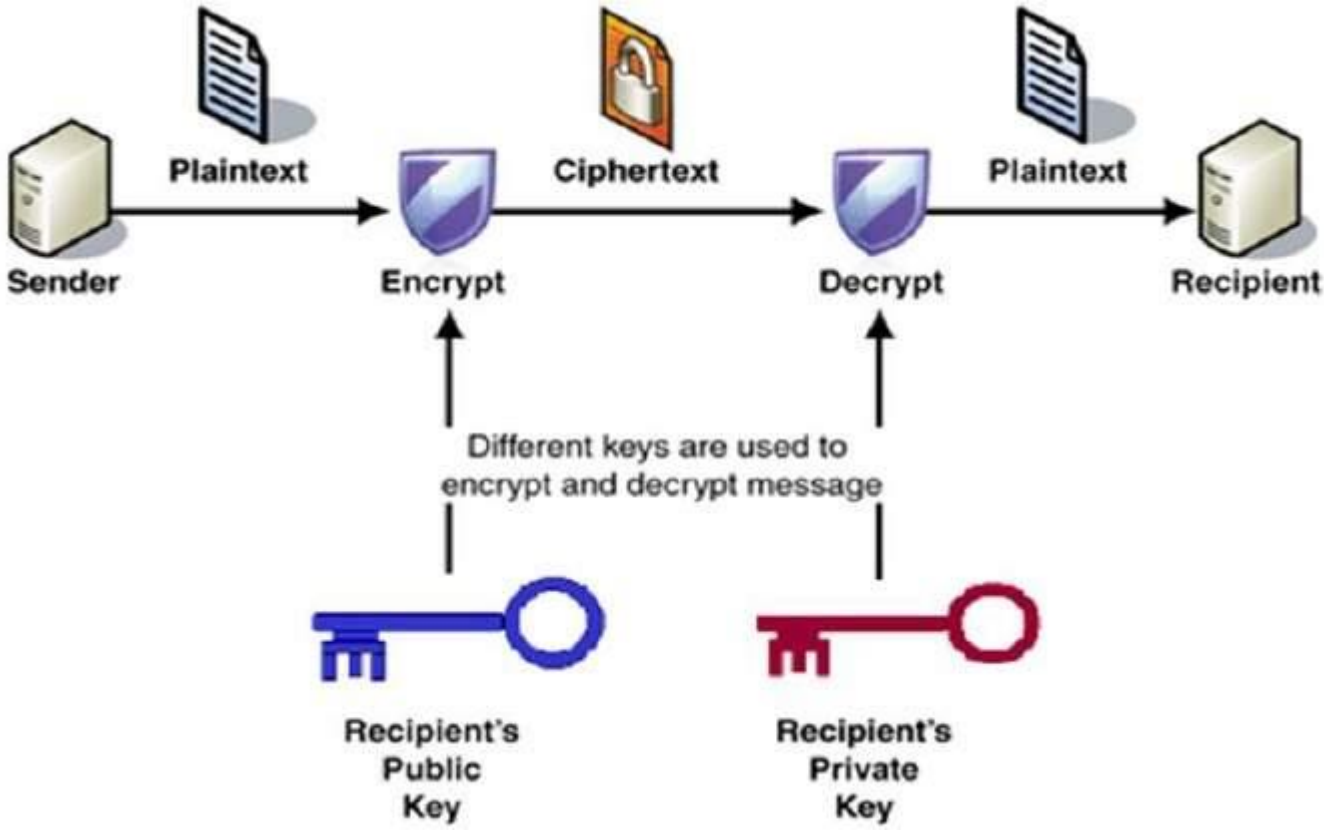


Fig.5.5 Encryption And Decryption Process

The most important properties of public key encryption scheme are −

- Different keys are used for encryption and decryption. This is a property which set this scheme different than symmetric encryption scheme.
- Each receiver possesses a unique decryption key, generally referred to as his private key.
- Receiver needs to publish an encryption key, referred to as his public key.
- Some assurance of the authenticity of a public key is needed in this scheme to avoid spoofing by adversary as the receiver. Generally, this type of cryptosystem involves trusted third party which certifies that a particular public key belongs to a specific person or entity only.
- Encryption algorithm is complex enough to prohibit attacker from deducing the plaintext from the ciphertext and the encryption (public) key.

- Though private and public keys are related mathematically, it is not be feasible to calculate the private key from the public key. In fact, intelligent part of any public-key cryptosystem is in designing a relationship between two keys.

There are three types of Public Key Encryption schemes. We discuss them in following sections −

RSA Cryptosystem

This cryptosystem is one the initial system. It remains most employed cryptosystem even today. The system was invented by three scholars **Ron Rivest, Adi Shamir,** and **Len Adleman** and hence, it is termed as RSA cryptosystem.

We will see two aspects of the RSA cryptosystem, firstly generation of key pair and secondly encryption-decryption algorithms.

Generation of RSA Key Pair

Each person or a party who desires to participate in communication using encryption needs to generate a pair of keys, namely public key and private key. The process followed in the generation of keys is described below −

- **Generate the RSA modulus (n)**
    - Select two large primes, p and q.
    - Calculate n=p*q. For strong unbreakable encryption, let n be a large number, typically a minimum of 512 bits.
- **Find Derived Number (e)**
    - Number **e** must be greater than 1 and less than (p − 1)(q − 1).
    - There must be no common factor for e and (p − 1)(q − 1) except for 1. In other words two numbers e and (p – 1)(q – 1) are coprime.
- **Form the public key**
    - The pair of numbers (n, e) form the RSA public key and is made public.
    - Interestingly, though n is part of the public key, difficulty in factorizing a large prime number ensures that attacker cannot find in finite time the two primes (p & q) used to obtain n. This is strength of RSA.

- **Generate the private key**
    - Private Key d is calculated from p, q, and e. For given n and e, there is unique number d.
    - Number d is the inverse of e modulo (p - 1)(q – 1). This means that d is the number less than (p - 1)(q - 1) such that when multiplied by e, it is equal to 1 modulo (p - 1)(q - 1).
    - This relationship is written mathematically as follows −

$$ed = 1 \bmod (p - 1)(q - 1)$$

The Extended Euclidean Algorithm takes p, q, and e as input and gives d as output.

Example

An example of generating RSA Key pair is given below. (For ease of understanding, the primes p & q taken here are small values. Practically, these values are very high).

- Let two primes be p = 7 and q = 13. Thus, modulus n = pq = 7 x 13 = 91.
- Select e = 5, which is a valid choice since there is no number that is common factor of 5 and $(p - 1)(q - 1) = 6 \times 12 = 72$, except for 1.
- The pair of numbers (n, e) = (91, 5) forms the public key and can be made available to anyone whom we wish to be able to send us encrypted messages.
- Input p = 7, q = 13, and e = 5 to the Extended Euclidean Algorithm. The output will be d = 29.
- Check that the d calculated is correct by computing −

$$de = 29 \times 5 = 145 = 1 \bmod 72$$

- Hence, public key is (91, 5) and private keys is (91, 29).

Encryption and Decryption

Once the key pair has been generated, the process of encryption and decryption are relatively straightforward and computationally easy.

Interestingly, RSA does not directly operate on strings of bits as in case of symmetric key encryption. It operates on numbers modulo n. Hence, it is necessary to represent the plaintext as a series of numbers less than n.

RSA Encryption

- Suppose the sender wish to send some text message to someone whose public key is (n, e).
- The sender then represents the plaintext as a series of numbers less than n.
- To encrypt the first plaintext P, which is a number modulo n. The encryption process is simple mathematical step as −

$$C = P^e \bmod n$$

- In other words, the ciphertext C is equal to the plaintext P multiplied by itself e times and then reduced modulo n. This means that C is also a number less than n.
- Returning to our Key Generation example with plaintext P = 10, we get ciphertext C

$$C = 10^5 \bmod 91$$

RSA Decryption

- The decryption process for RSA is also very straightforward. Suppose that the receiver of public-key pair (n, e) has received a ciphertext C.
- Receiver raises C to the power of his private key d. The result modulo n will be the plaintext P.

$$\text{Plaintext} = C^d \bmod n$$

- Returning again to our numerical example, the ciphertext C = 82 would get decrypted to number 10 using private key 29

$$\text{Plaintext} = 82^{29} \bmod 91 = 10$$

RSA Analysis

The security of RSA depends on the strengths of two separate functions. The RSA cryptosystem is most popular public-key cryptosystem strength of which is based on the practical difficulty of factoring the very large numbers.

- **Encryption Function** − It is considered as a one-way function of converting plaintext into ciphertext and it can be reversed only with the knowledge of private key d.

- **Key Generation** − The difficulty of determining a private key from an RSA public key is equivalent to factoring the modulus n. An attacker thus cannot use knowledge of an RSA public key to determine an RSA private key unless he can factor n. It is also a one way function, going from p & q values to modulus n is easy but reverse is not possible.

If either of these two functions are proved non one-way, then RSA will be broken. In fact, if a technique for factoring efficiently is developed then RSA will no longer be safe.

The strength of RSA encryption drastically goes down against attacks if the number p and q are not large primes and/ or chosen public key e is a small number.

ElGamal Cryptosystem

Along with RSA, there are other public-key cryptosystems proposed. Many of them are based on different versions of the Discrete Logarithm Problem.

ElGamal cryptosystem, called Elliptic Curve Variant, is based on the Discrete Logarithm Problem. It derives the strength from the assumption that the discrete logarithms cannot be found in practical time frame for a given number, while the inverse operation of the power can be computed efficiently.

Let us go through a simple version of ElGamal that works with numbers modulo p. In the case of elliptic curve variants, it is based on quite different number systems.

Generation of ElGamal Key Pair

Each user of ElGamal cryptosystem generates the key pair through as follows −

- **Choosing a large prime p.** Generally a prime number of 1024 to 2048 bits length is chosen.
- **Choosing a generator element g.**
    - This number must be between 1 and p − 1, but cannot be any number.
    - It is a generator of the multiplicative group of integers modulo p. This means for every integer m co-prime to p, there is an integer k such that g^k=a mod n.

For example, 3 is generator of group 5 ($Z_5 = \{1, 2, 3, 4\}$)

N	3^n	$3^n \bmod 5$
1	3	3
2	9	4
3	27	2
4	81	1

- **Choosing the private key.** The private key x is any number bigger than 1 and smaller than p−1.
- **Computing part of the public key.** The value y is computed from the parameters p, g and the private key x as follows −

$$y = g^x \bmod p$$

- **Obtaining Public key.** The ElGamal public key consists of the three parameters (p, g, y).

  For example, suppose that p = 17 and that g = 6 (It can be confirmed that 6 is a generator of group Z_{17}). The private key x can be any number bigger than 1 and smaller than 71, so we choose x = 5. The value y is then computed as follows −

$$y = 6^5 \bmod 17 = 7$$

- Thus the private key is 62 and the public key is (17, 6, 7).

Encryption and Decryption

The generation of an ElGamal key pair is comparatively simpler than the equivalent process for RSA. But the encryption and decryption are slightly more complex than RSA.

ElGamal Encryption

Suppose sender wishes to send a plaintext to someone whose ElGamal public key is (p, g, y), then −

- Sender represents the plaintext as a series of numbers modulo p.
- To encrypt the first plaintext P, which is represented as a number modulo p. The encryption process to obtain the ciphertext C is as follows −
    - Randomly generate a number k;
    - Compute two values C1 and C2, where −

$$C1 = g^k \bmod p$$
$$C2 = (P^* y^k) \bmod p$$

- Send the ciphertext C, consisting of the two separate values (C1, C2), sent together.
- Referring to our ElGamal key generation example given above, the plaintext P = 13 is encrypted as follows −
    - Randomly generate a number, say k = 10
    - Compute the two values C1 and C2, where −

$$C1 = 6^{10} \bmod 17$$
$$C2 = (13^* 7^{10}) \bmod 17 = 9$$

- Send the ciphertext C = (C1, C2) = (15, 9).

ElGamal Decryption

- To decrypt the ciphertext (C1, C2) using private key x, the following two steps are taken −
    - Compute the modular inverse of $(C1)^x$ modulo p, which is $(C1)^{-x}$, generally referred to as decryption factor.

- Obtain the plaintext by using the following formula −

$$C2 \times (C1)^{-x} \bmod p = \text{Plaintext}$$

- In our example, to decrypt the ciphertext C = (C1, C2) = (15, 9) using private key x = 5, the decryption factor is

$$15^{-5} \bmod 17 = 9$$

- Extract plaintext P = (9 × 9) mod 17 = 13.

ElGamal Analysis

In ElGamal system, each user has a private key x. and has **three components**of public key − **prime modulus p, generator g, and public Y = g^x mod p**. The strength of the ElGamal is based on the difficulty of discrete logarithm problem.

The secure key size is generally > 1024 bits. Today even 2048 bits long key are used. On the processing speed front, Elgamal is quite slow, it is used mainly for key authentication protocols. Due to higher processing efficiency, Elliptic Curve variants of ElGamal are becoming increasingly popular.

Elliptic Curve Cryptography (ECC)

Elliptic Curve Cryptography (ECC) is a term used to describe a suite of cryptographic tools and protocols whose security is based on special versions of the discrete logarithm problem. It does not use numbers modulo p.

ECC is based on sets of numbers that are associated with mathematical objects called elliptic curves. There are rules for adding and computing multiples of these numbers, just as there are for numbers modulo p.

The shorter keys result in two benefits −

- Ease of key management
- Efficient computation

These benefits make elliptic-curve-based variants of encryption scheme highly attractive for application where computing resources are constrained.

RSA and ElGamal Schemes – A Comparison

Let us briefly compare the RSA and ElGamal schemes on the various aspects.

RSA	ElGamal
It is more efficient for encryption.	It is more efficient for decryption.
It is less efficient for decryption.	It is more efficient for decryption.
For a particular security level, lengthy keys are required in RSA.	For the same level of security, very short keys are required.

Table 5.1 Comparison of RSA and ElGamal

5.4 Introduction to AES Algorithm:

Advanced Encryption Standard, also known as the Rijndael (pronounced as Rain Doll) algorithm is adopted worldwide.AES Algorithm is used to protect Electronic data. The first thing AES Algorithm needs is data as input and the other thing it needs is key(encryption key). When these two combined are called as input and are feed into Cipher Engine produces Encrypted data in binary format called as cipher text. To recover the encrypted data it has to reverse the process in which the cipher text and key is feed into Cipher Engine to get back the original data. AES is a symmetric block cipher that can encrypt data blocks of 128 bits using symmetric keys.

Rounds:

There are 10, 12, 14 rounds for 128.192 and 256 bit keys. Regular rounds are 9,11and 13.Final round is 10th, 12th, 14th. Each round has certain processing involved. Following are the transformation involved.

1.SubBytes Transformation:-It uses substitution table which includes nonlinear substitution which operate on each byte of the state.

2.ShiftRows Transformation:- In ShiftRows step, bytes in each row of the state are shifted cyclically to the left. The number of places each byte is shifted differs for each row. The first row doesn’t change.

3.MixColumnsTransformation:-MixColumns step operates on the column level. It is equivalent to the multiplication of matrix at column level. Each column of the state I multiplied with fixed polynomial.

4.AddRoundKeyTransformation:-In dd RoundKey step, the state is combined with roundkey using XOR operation.

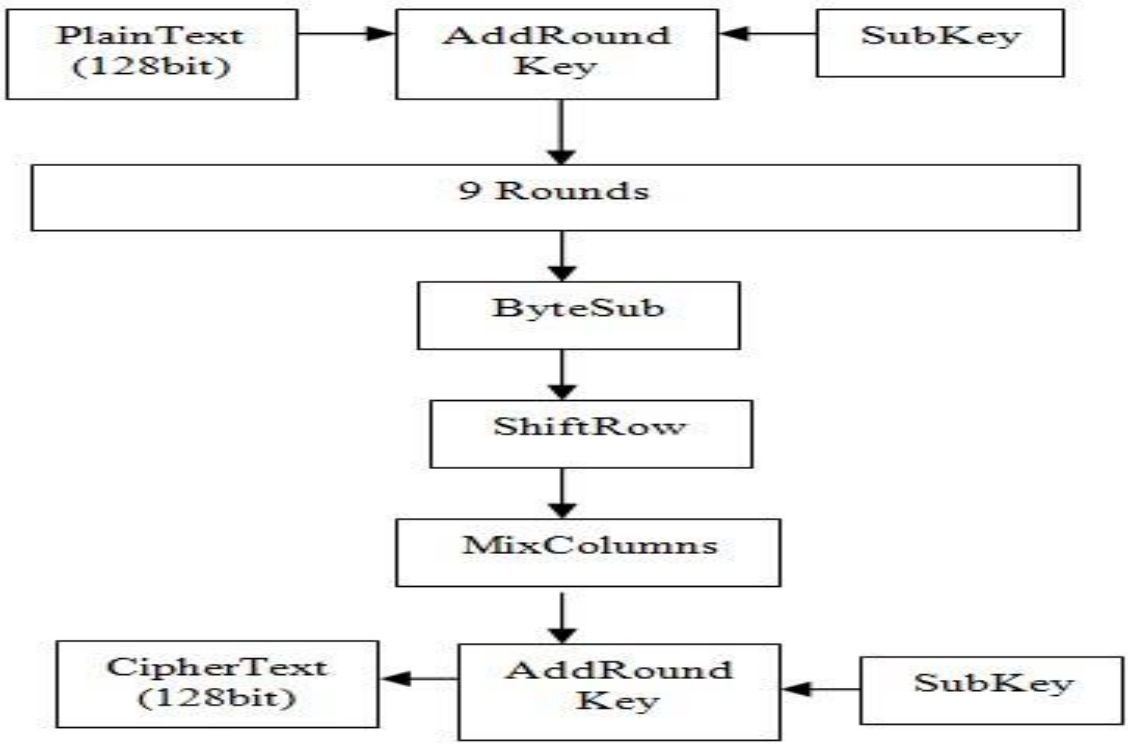


Fig. 5.6 Algorithm Encryption structure

Expansion Key:-In AES algorithm, the sender and receiver is known about the key. The AES algorithm remains secure, the key cannot be determined any intruder even if he knows the plaintext and the cipher text. The AES algorithm is designed to use one of three key sizes (Nk).The keys can be 128 bits, 192 bits, 256 bits.128 bits means (16 bytes, 4 words) ,192 .3bits means (24 bytes, 6 words), 256 bits means (32 bytes, 8 words). These are the key sizes which are supported by AES Encryption. The larger the key the stronger is the encryption.The keys are then expanded via a key expansion routine for use in the AES cipher algorithm.

5.5 Basic principles of the IWD algorithm

Water drops that flow in rivers, lakes, and seas are the sources of inspiration for developing the IWD. This intelligence is more obvious in rivers which find their ways to lakes, seas, or oceans despite many different kinds of obstacles on their ways. In the water drops of a river, the gravitational force of the earth provides the tendency for flowing toward the destination. If there were no obstacles or barriers, the water drops would follow a straight path toward the destination, which is the shortest path from the source to the destination. However, due to different kinds of obstacles in their way to the destination, which constrain the path construction, the real path has to be different from the ideal path and lots of twists and turns in the river path is observed. The interesting point is that this constructed path seems to be optimum in terms of distance from the destination and the constraints of the environment.

CHAPTER 6

Result & Result Analysis

6.1 Introduction

Image is encrypted and decrypted using AES Algorithm. encryption quality. Even AES-128 offers a sufficiently large number of possible keys, making an exhaustive search impractical for many decades Encryption and decryptio encryption by AES Algorithm is less than the time required by DES Algorithm. the algorithm is suitable for image encryption in real time applications.

6.2 Technical Contribution

It was first proposed for authentication and its important feature is reversibility, it hide the secret data in the digital image in such a way that only the authorized person could decrypt the secret information and restore the original image. Several data hiding methods have been proposed. The performance of a reversible data embedding algorithm is measured by its payload capacity, complexity, visual quality and security. Earlier methods have lower embedding capacity and poor image quality. As the embedding capacity and image quality is improved, this method became a convert communication channel. Not only the data hiding algorithm be given importance but also the image on which the data is hidden should also be highly secured. This security is provided in two layers. First the data that is to be hidden in the image is encrypted using AES algorithm. This encrypted data is then hidden in the image. The image with the hidden data is then encrypted again. Thus the user with the decryption key of both image and data will be able to retrieve the data and image in its original form.

6.2.1 Strength and Efficiency of the Technique

AES is a symmetric key block cipher published by the NIST in December 2001. NIST evaluation criteria for AES are Security Cost Algorithm and Implementation Characteristics. AES is a non-Feistel cipher that encrypts and decrypts a data block of 128-bits. The key size can be 128,192 or 256-bits. It depends on number of rounds. The input to the encryption and decryption algorithm is a single 128-bit block. The block is represented as a row of matrix of 16 bytes. AES structure is not a Feistel structure. Encryption is the process of converting plaintext to cipher-text (had to understand) by applying mathematical transformations. These transformations are known as encryption

algorithms and require an encryption key. Decryption is the reverse process of getting back the original data from the cipher-text using a decryption key. In Symmetric cryptology- The encryption key and the decryption key could be the same as in symmetric or secret key cryptography, The key can different as in asymmetric or public key cryptography

6.3 Economical Contribution of the Research

The distributed source coding (DSC) to encrypt image in RDH, by encrypting the original image/media using stream cipher, the data-hider compresses a series of selected bits which is taken from the encrypted image to make the secret data. The original image is encrypted directly by the sender and the data-hider embeds the additional bits by modifying some bits of the encrypted data. Data extraction and image recovery are realized by analyzing the local standard deviation during decryption of the marked encrypted image. The receiver end has both the embedding and encryption key and then the receiver can extract the secret data and recover the original image perfectly using the distributed source decoder. The expected result is lossless image and data. Thus in our project the security of the image can be enhanced by encrypting the data and the image in which the data is hidden. The receiver should have three keys to retrieve the data (i.e.) the decryption key of the data, the retrieving key of the data from the image and lastly the decrypting key of the image.

6.3.1 Knowledge Gain

The existing system uses the histogram of the image to embed the data. This method also enhances the contrast of the image. The image enhancement is achieved by histogram equalisation. The highest peaks in the histogram are taken. The bins between the peaks are unchanged while the outer bins are shifted outwards so that each of the two peaks can be split into two adjacent bins. To increase embedding capacity, the highest two bins in the modified histogram can be further chosen to be split, and so on until satisfactory construct enhancement effect is achieved. For the recovery of the original image, the location map is embedded into the host image, together with the message bits and other side information. So blind data extraction and complete recovery of the original image are both enabled. The generation of image histogram is a difficult and a time consuming process. But the contrast of the image is enhanced. The data is only hidden in the image where the security level is simple. Since the data is hidden and if the retrieving process is known the intruder will be able to retrieve the image easily without any effort**.**

6.3.2 Security Enhancement

Data Security is primary concern for every communication system. The relentless growth of Internet and communication technologies has made the extensive use of images unavoidable. There are many ways to provide security to data that is being communicated. This Paper describes a design of effective security for communication by AES algorithm for encryption and decryption. It is based on AES Key Expansion in which the encryption process is a bit wise exclusive or operation of a set of image pixels along with the a 128 bit key which changes for every set of pixels. The National Institute of Standards and Technology (NIST) has initiated a process to develop a Federal information Processing Standard (FIPS) for the Advanced Encryption Standard (AES), specifying an Advanced Encryption Algorithm to replace the Data Encryption standard (DES) the Expired in 1998. The Advanced Encryption Standard can be programmed in software or built with pure hardware**.**

6.4. Random Factor

Using these algorithms allow separately kind of luxurious ensure confidentiality. For this reason, a hybrid cryptosystem based on both AES and RSA is proposed. The Advanced Encryption Standard (AES) and the Rivest Shamir Adleman (RSA) algorithms are the two popular encryption algorithms that vouch confidentiality, integrity and authenticity over an insecure communication network and Internet.AES algorithm which contain iterative rounds. AES algorithm support several cipher modes of operation such as ECB (Electronic Code Book), CBC (Cipher Block chaining), OFB (Output Feedback), CFB (Cipher Feedback) and CTR (Counter) [13-15]. In our system, privacy is ensured by AES algorithm using five modes of operation and the RSA algorithm is used to transmit the keys. The cryptosystem also check the integrity of images using a simple process based on correlation between the pixels of Meteosat images. The rest of the paper is organized as follow. Section 2 discusses the proposed hybrid cryptosystem scheme. Section 3 and 4 shows some numerical results.

6.5 Results:

The proposed algorithm is implemented on Matlab. After starting Matlab click on open and select the folder having the project on computer or any storage. In this preject we are doing Encryption as well as Decryption both so for Encrytion we have to select a picture. Suppose we selected a pic of audi from my computer it can be of any format.

Figure 6.1 Input Image

Now it will ask for a key for Encryption. As we are using symmetric key for encryption so that we have to remember the key. When we enter the key it will start encrypting the image. Second picture which will appear on the screen that will look like this

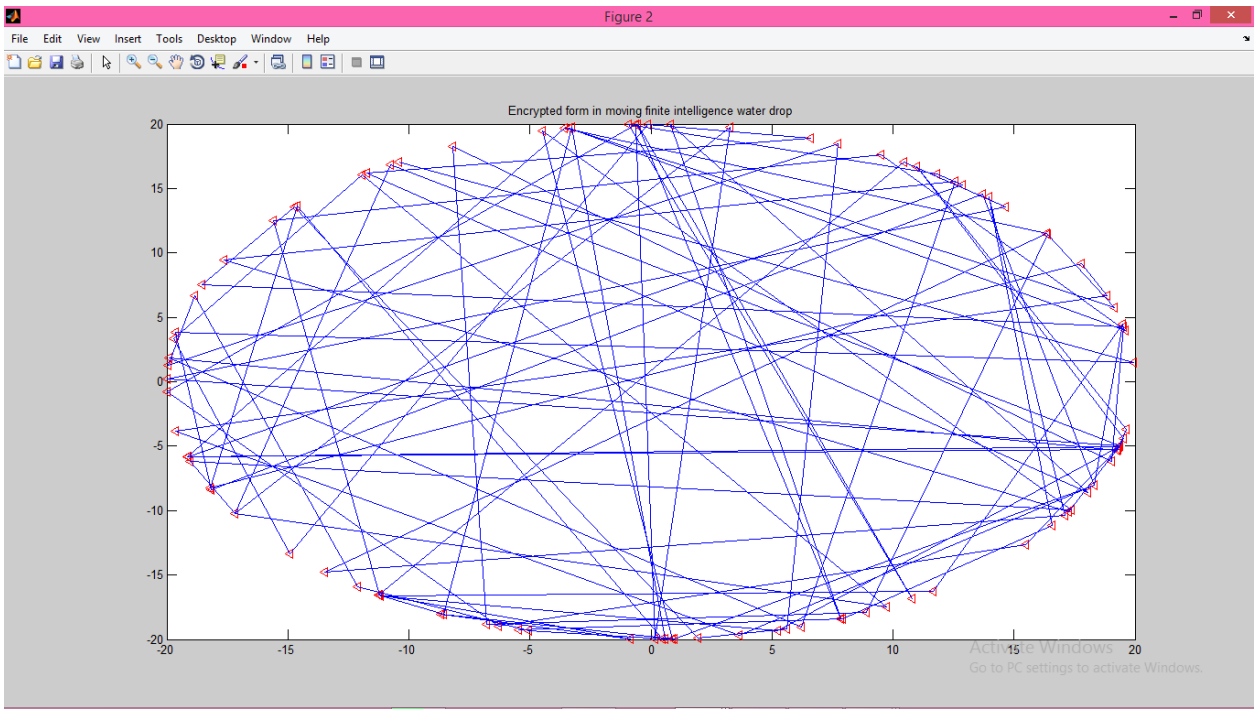


Figure 6.2 Path of Encrypted Image

This is Moving pixel path of Encrypted Image which is generated by IWD algorithm. To check the path followed by IWD for the successful transaction of imaginary data can be generated. And then the next picture will be Encrypted Image of that image

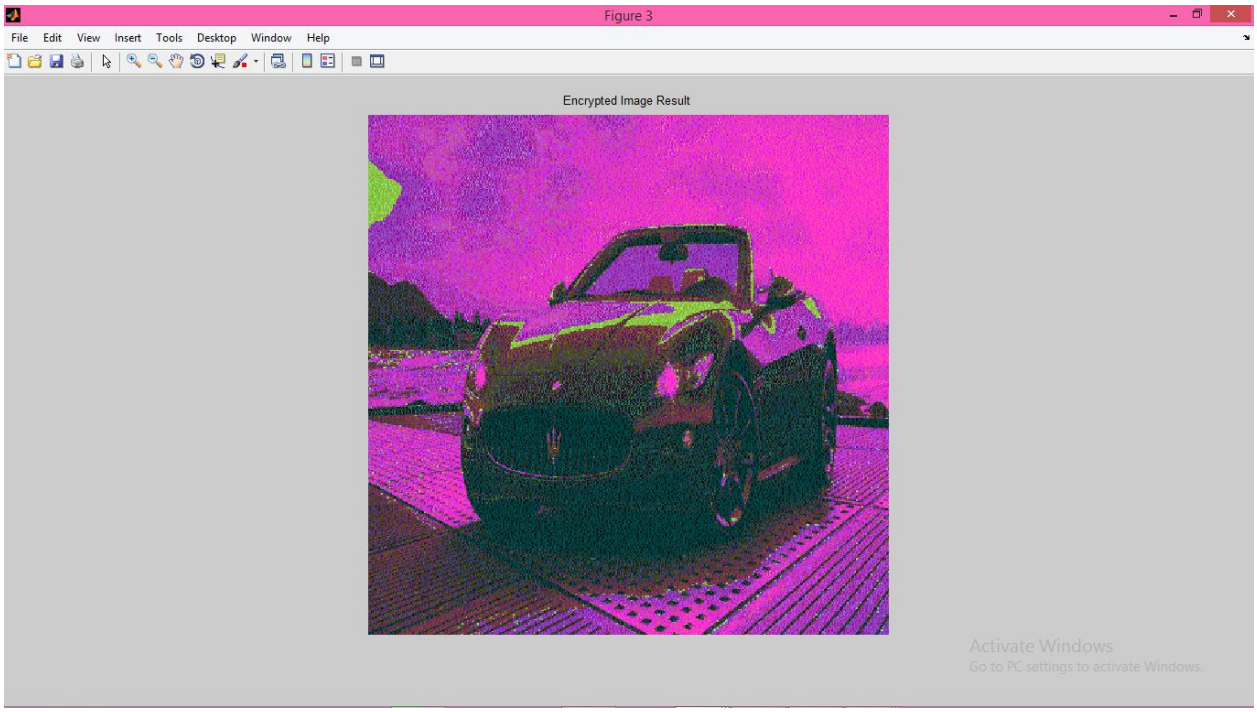


Figure 6.3 Encrypted Image

Then It will Ask a key for Decryption. And we have to enter that same key which we entered for encryption if both keys mismatch then it will show a message for wrong key As shown below.

Figure 6.4 For Wrong Input

If you Enter the correct key then it will Decrypt that encrypted image and we can see final image which will look same.

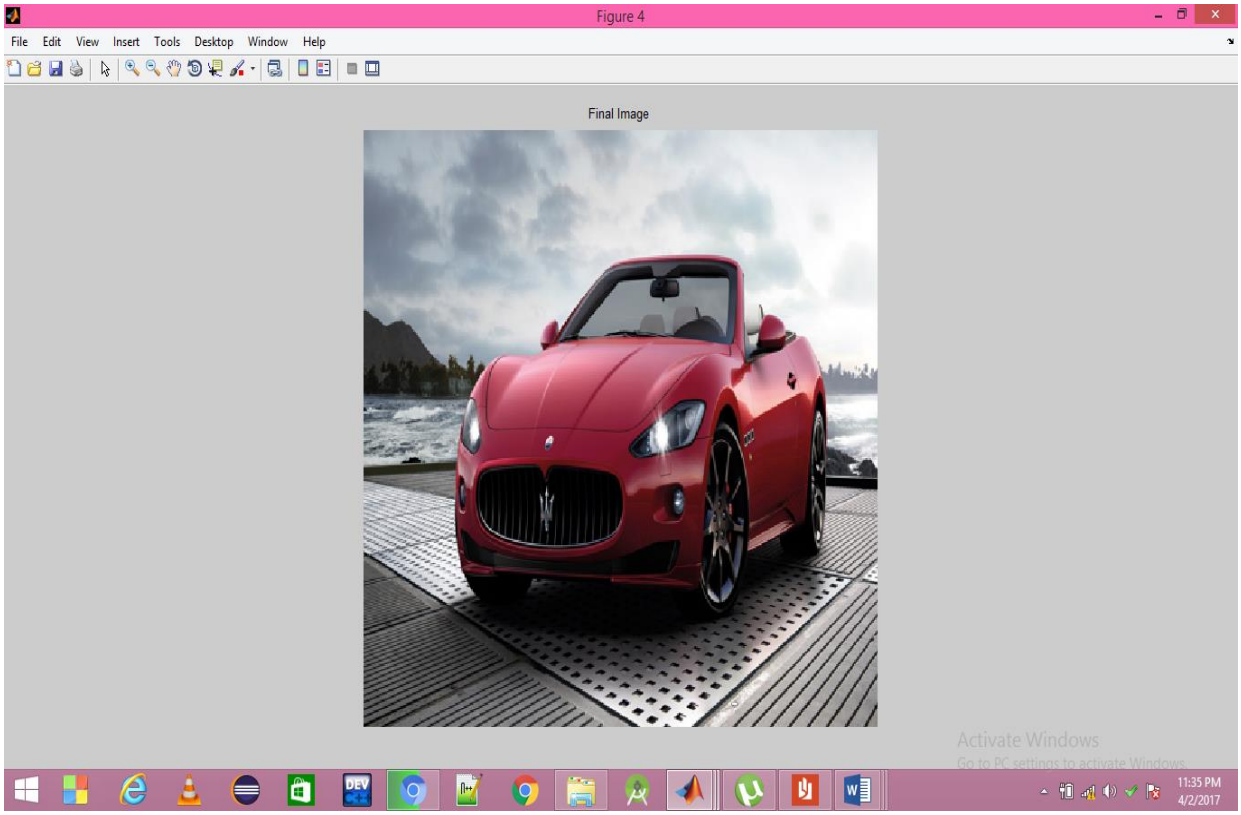


Fig6.5 Final Image

And some Histograms as Histogram for Image And Encrypted image and then final Image Histogram. Histogram of input Image will be quiet similar to the final image. All Histograms are shown below

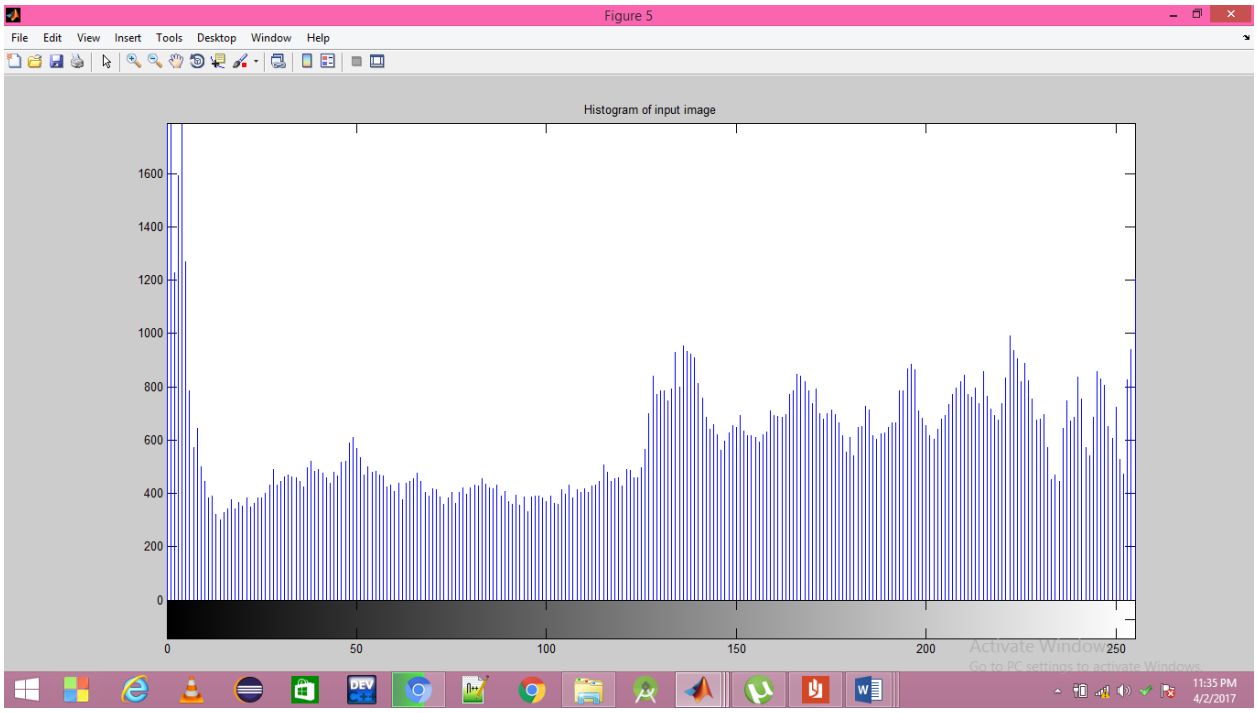


Figure 6.6 Histogram for input Image

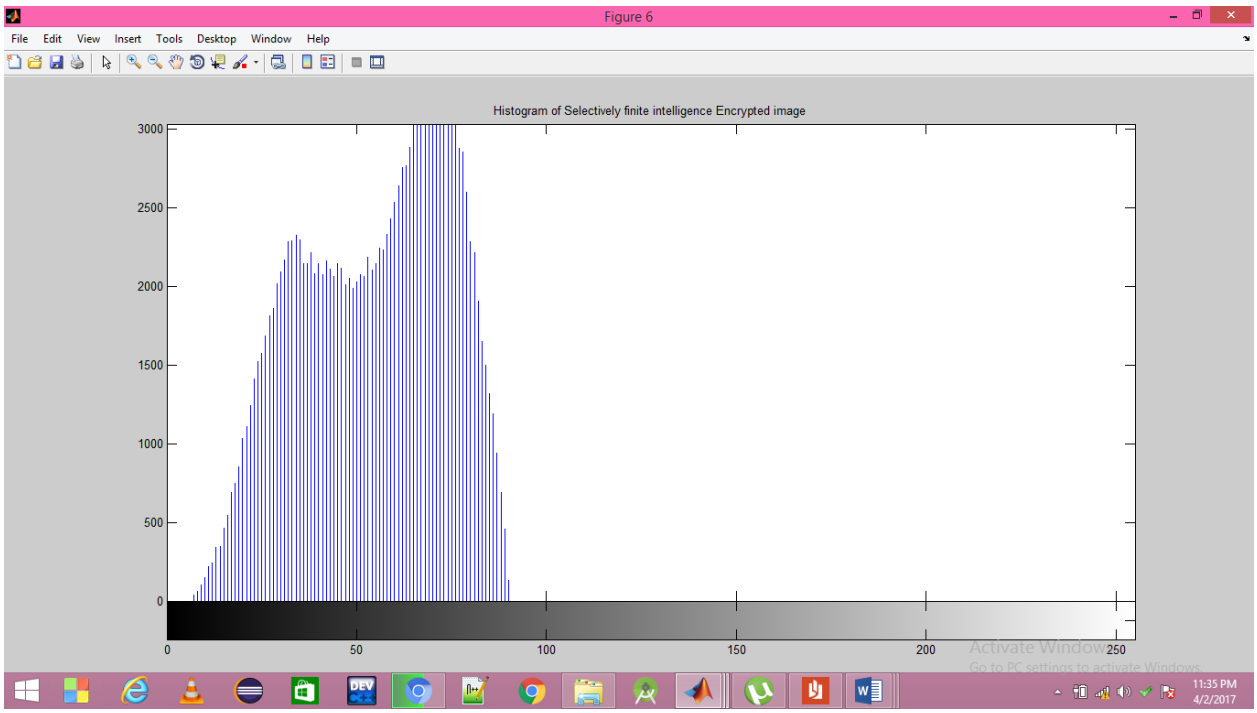


Figure 6.7 Histogram for Encrypted Image

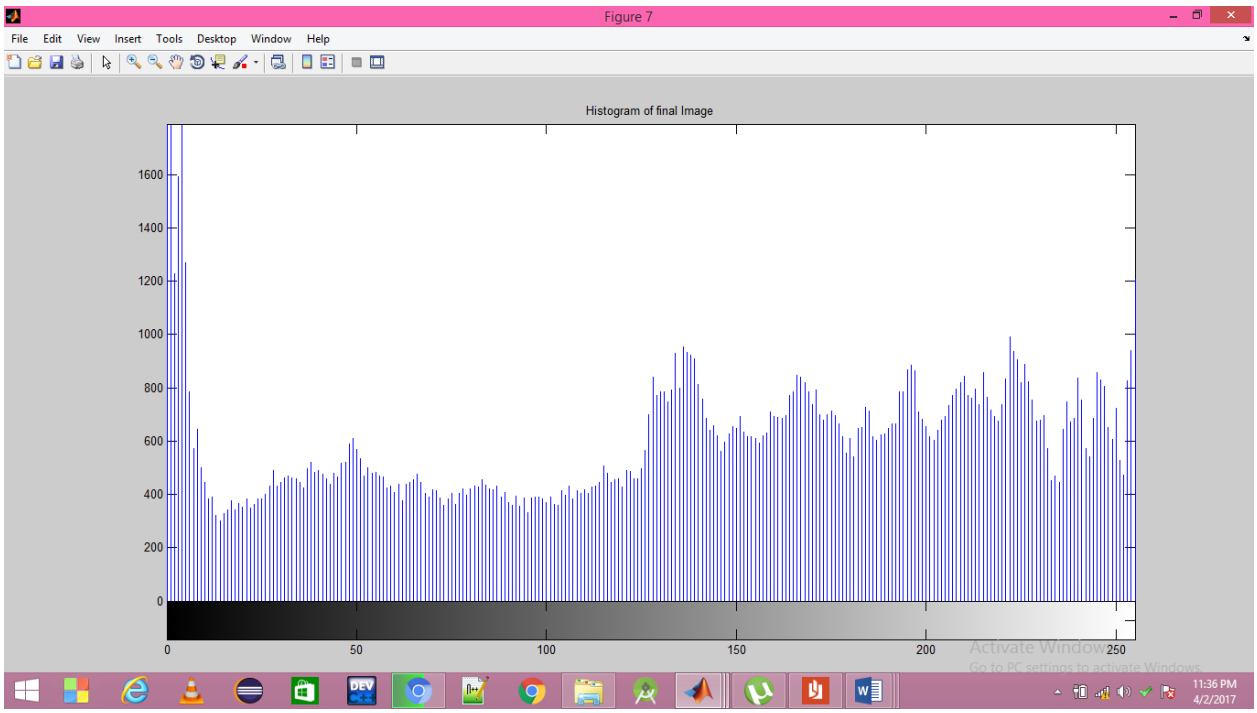


Figure 6.8 Histogram of Final Image

Just after the Histograms there will appear graph which is ploted for both Encryption an Decryption both. And this show performance time

1

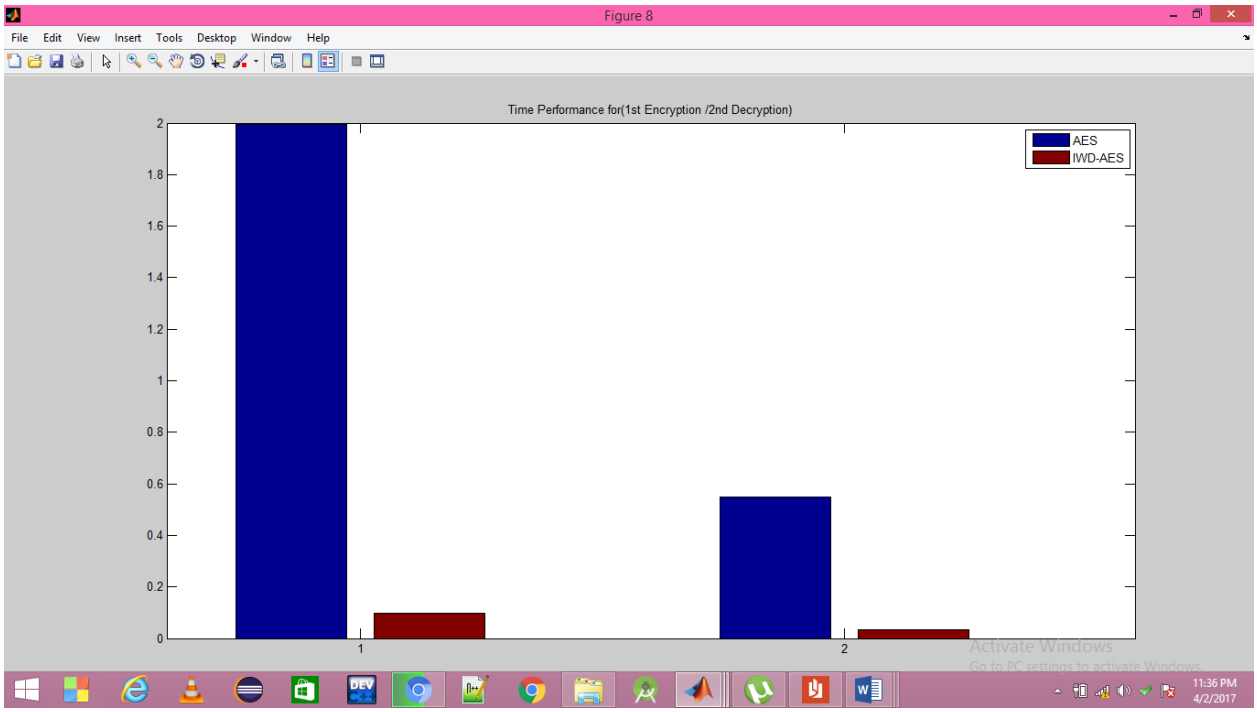


Figure 6.9 Graph of Performance Time

Image	**Size of image(kb)**	**Encryption time using AES only(ms)**	**Encryption time using AES & IWD(ms)**
jellyfish	758	1.82	.2a
penguins	760	1.83	.14
Audi	687	1.9	.09

Table 6.1

Image size (in pixels)	**Image size (kb)**	**Decryption time using AES only(ms)**	**Decryption time using AES & IWD(ms)**
jellyfish	758	0.59	0.05
penguins	760	0.6	0.04
audi	687	0.56	0.05

Table 6.2

CHAPTER 7

CONCLUSION & FUTURE SCOPE

The difference of efficiency between our “Proposed Algorithm” and “Image Encryption Using Block-Based Transformation Algorithm”, “An Image Encryption Approach Using a Combination of Permutation Technique Followed by Encryption” is very high approximately 80% . If the security and efficiency is of primary concern then one can use our proposed algorithm. From the above discussion we can clearly see that the proposed algorithm has 70% better entropy of encrypted image any of the other compeering algorithms and hence can be incorporated in the process of encryption of any images. Also, we can see that the “Image Encryption Using BlockBased Transformation Algorithm” and “An Image Encryption Approach Using a Combination of Permutation Technique Followed by Encryption” have very less entropy and hence cannot be used for encryption of confidential messages. As the embedding capacity and image quality is improved, this method became a convert communication channel. Not only the data hiding algorithm be given importance but also the image on which the data is hidden should also be highly secured. This security is provided in two layers. First the data that is to be hidden in the image is encrypted using AES algorithm. This encrypted data is then hidden in the image. The image with the hidden data is then encrypted again. Thus the user with the decryption key of both image and data will be able to retrieve the data and image in its original form. The encryption algorithm presented above, is a very simple, direct mapping algorithm using feistal Structure and some logical operation. This cipher image generation provides a good strength to the encryption algorithm. As such it is quite essential to improve our algorithms performance in future.

Future Work

The difference of efficiency between our “Proposed Algorithm” and “Image Encryption Using Block-Based Transformation Algorithm”, “An Image Encryption Approach Using a Combination of Permutation Technique Followed by Encryption” is very high approximately 80% . If the security and efficiency is of primary concern then one can use our proposed algorithm. From the above discussion we can clearly see that the proposed algorithm has 70% better entropy of encrypted image any of the other compeering algorithms and hence can be incorporated in the process of encryption of any images.

We can see that the “Image Encryption Using Block-Based Transformation Algorithm” and “An Image Encryption Approach Using a Combination of Permutation Technique Followed by Encryption” have very less entropy and hence cannot be used for encryption of confidential messages. The encryption algorithm presented above, is a very simple, direct mapping algorithm using feistal Structure and some logical operation. This cipher image generation provides a good strength to the encryption algorithm. As such it is quite essential to improve our algorithms performance in future. Some Future works are given as following:

→**Image enhancing:** The camera apps in smartphones and digital cameras using image processing to enhance the image quality, video stabilization and noise removal etc

→**Gaming:** Advanced gaming consoles like Xbox kinect uses image processing from motion analysis of the human player.

→**Human machine interface:** machines are made smart by adding gestural interface, or human action response interfaces, which decodes the actions of the human user to perform certain tasks.

→**Problem specific solutions:** image processing is used as a solution to a variety of problems, starting from facial recognition access to defects identification in manufacturing industries

REFERENCES

[1]. Abdelfatah A. Yahya and Ayman M. Abdalla"A Shuffle Image-Encryption Algorithm"Department of Computer Science, Al-Zaytoonah University of Jordan, Journal of Computer Science 4

[2]. Bruce schneier"Applied Cryptography" 2nd Edition published by John Wiley&SonsInc.

[3]. Diffie, W., & Hellman, M. E. New directions in cryptography. Information Theory, IEEE Transactions

[4]. Guan, Z. H., Huang, F., & Guan, W.Chaosbased image encryption algorithm. Physics Letters A

[5]. Hatamlou, A. (2013). Black hole: A new heuristic optimization approach for data clustering. Information Sciences,.

[6] J. Daemen, L. R. Knudsen, and V. Rijmen: The Galois Field GF(28). http: //www.ddj.com/documents/s=936/ddj9710e/9710es1.htm, Dr. Dobb's Journal.

[7] J. Daemen, V. Rijmen: AES proposal: Rijndael. http://www.esat.kuleuven.ac. be/~rijmen/rijndael/rijndaeldocV2.zip.

[8]. Kamkar, I., Akbarzadeh-T, M. R., & Yaghoobi, M. (2010, October). Intelligent water drops a new optimization algorithm for solving the vehicle routing problem. In Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on. IEEE.

[9]. Madan, B. B., Goševa-Popstojanova, K., Vaidyanathan, K., & Trivedi, K. S. A method for modeling and quantifying the security attributes of intrusion tolerant systems. Performance Evaluation,.

[10]. M. Zeghid, M. Machhout, L. Khriji, A. Baganne, and R. Tourki "A Modified AES Based Algorithm for ImageEncryption" World Academy of Science, Engineering and Technology 27, 2007

[11]. Salmanpour, S., Omranpour, H., & Motameni, H. (2013, November). An intelligent water drops algorithm for solving robot path planning problem. In Computational Intelligence and Informatics (CINTI), 2013 IEEE 14th International Symposium on IEEE.

[12]. Shah-Hosseini, H. Problem solving by intelligent water drops. In Evolutionary Computation, 2007. CEC 2007. IEEE Congress. Shah-Hosseini, H. (2009). The intelligent water drops algorithm: a nature-inspired swarm-based optimization algorithm. International Journal of Bio-Inspired Computation.

[13]. Shah-Hosseini, H. (2009). The intelligent water drops algorithm: a nature-inspired swarm-based optimization algorithm. International Journal of Bio-Inspired Computation.

[14]. Shah-Hosseini, H. (2008). Intelligent water drops algorithm: A new optimization method for solving the multiple knapsack problem. International Journal of Intelligent Computing and Cybernetics, 1(2).

[15]. Shtewi, A. A., Hasan, B. E. M., & Hegazy, A. E. F. (2010). An efficient modified advanced encryption standard (MAES) adapted for image cryptosystems. IJCSNS International Journal of Computer Science and Network Security,.

[16] The Mathworks: Matlab , The Language of Technical Computing. http://www. mathworks.com/products/matlab, .

[17] The Mathworks: Galois Field Computations. http://www.mathworks.com/ access/helpdesk/help/toolbox/comm/tutor3.shtml, Communications Toolbox.

[18] P.Karthigaikumar Simulation of Image Encryption using AES Algorithm

[19]. William, S., & Stallings Cryptography and Network Security, 4/E. Pearson Education India.

[20]. William stallings "Cryptography and Network Security" 3rd Edition published by Pearson Education Inc and Dorling Kindersley Publishing Inc.

[21]. Xinmiao Zhang, Student Member,IEEE, and Keshab K. Parthi, Fellow, IEEE "High-Speed VLSI Architecture for AES Algorithm" IEEE Transactions on VLSI,