SECURITY EXTENSIBILITY IN DATA TRANSFER COMBINING CRYPTOGRAPHY AND STEGANOGRAPHY

Project Report submitted in partial fulfilment of the requirement for the degree of

Bachelor of Technology

in

ELECTRONICS AND COMMUNICATION ENGINEERING

under the supervision of

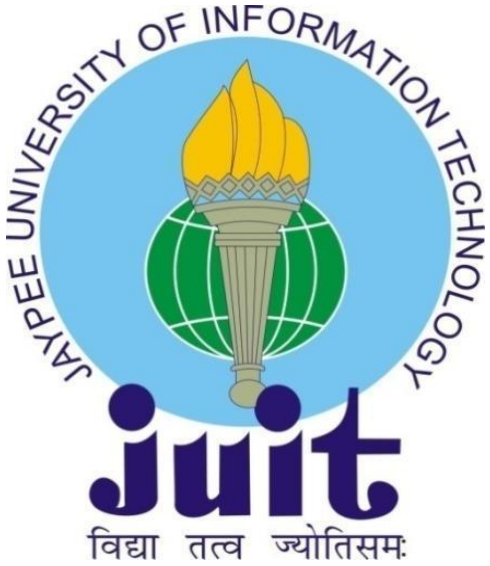
Ms. Meenakshi Sood

By

Nitika Rana (111065)

Shipra Sharma (111035)

to



JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY

WAKNAGHAT, SOLAN – 173234, HIMACHAL PRADESH

CERTIFICATE

This is to certify that the project report entitled "**Security extensibility in data transfer combining Cryptography and Steganography**", submitted by Nitika Rana and Shipra Sharma in partial fulfillment for the award of degree of Bachelor of Technology in Electronics and Communication Engineering to Jaypee University of Information Technology , Waknaghat, Solan has been carried out under my supervision.

This work has not been submitted partially or fully to any other university or institute for the award of this or any other degree or diploma.

Date :

Ms. Meenakshi Sood

Asst. Professor

Dept. of ECE

ACKNOWLEDGEMENT

The joy and sense of fulfillment that comes along with the successful completion of any task is incomplete without thanking all those people who made it possible with their guidance and constant word of encouragement.

We thank Ms. Meenakshi Sood (Asst. Professor, JUIT, H.P.) for her guidance and constant supervision as well as for providing necessary information regarding the project and also for her support in completing the project.

We thank out Dean, Prof. T. S. Lamba, who has always served as an inspiration for us. We would like to express sincere appreciation to our Head of Department, Prof. S. V. Bhooshan and the whole Electonics and Communication Department for having extended all the department facilities without hesitation.

We thank God Almighty for giving us such an excellent facilities and support through the way of JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, for giving us the opportunity and facilities for completion of this project.

Date: **Shipra Sharma (111035)**

Nitika Rana (111065)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The rapid development of data transfer through internet made it easier to send the data accurate and faster to the destination. There are many transmission media to transfer the data from source to destination. At the same time is easy to modify and misuse the valuable information through hacking.

Steganography is the art and science of writing hidden messages in such a way that no one apart from the sender and intended recipient even realizes there is a hidden message. There are often cases when it is not possible to send messages openly or in encrypted form. This is where Steganography can come into play. Steganography can be viewed as akin to cryptography. Both have been used throughout recorded history as means to protect information. While cryptography provides privacy, Steganography is intended to provide secrecy. Cryptographic cipher is used before hiding text to make this procedure more secure. Cryptographic technique is one of the principal means to protect information security. Not only has it to ensure the information confidential, but also provides digital signature, authentication, secret sub-storage, system security and other functions. Therefore, the encryption and decryption solution can ensure the confidentiality of the information, as well as the integrity of information and certainty, to prevent information from tampering, forgery and counterfeiting.

This project deals with hiding of encrypted text behind a digital image, to ensure both privacy and secrecy. Encryption is done with the help of RSA encrypt/decrypt algorithm, on the other hand for Steganography; we have proposed and implemented a new technique which is an enhancement to conventional LSB technique.

CHAPTER 1 INTRODUCTION

1.1 Overview

The rapid development of data transfer through internet made it easier to send the data accurate and faster to the destination. There are many transmission media to transfer the data from source to destination. At the same time is easy to modify and misuse the valuable information through hacking.

Information hiding is one of the important areas of information security, which includes various methods like cryptography, steganography and watermarking. In Cryptography encryption is done results in a disordered and perplexing message. Though the message cannot read by the third party but attract eavesdroppers easily. Steganography overcome this problem by hiding the secret information behind a cover media (video, audio or image) because the presence of information cannot be noticed by any attacker. The goal of steganography is to embed secret data into a cover in such a way that no one apart from the sender and intended recipients even realizes there is a secret data.

1.2 Objective:

The aim of the project is to implement an algorithm which increases security of the data we transfer through internet, so that it is immune to the attackers.

- To study difference between Cryptography and Steganography
- Converting plain text into cipher text.
- Embedding the cipher text into an Image.
- Implementing our own proposed algorithm for steganography.
- Comparing the results and errors of different algorithms used

1.3 Steganography and Cryptography

Steganography is the art and science of invisible communication. This is accomplished through hiding information in other information, thus hiding the existence of the communicated information. The word steganography is derived from the Greek words "*stegos*" meaning "cover" and "*grafia*" meaning "writing" defining it as "covered writing". In image steganography the information is hidden exclusively in images.

Cryptography is derived from Greek word kryptós means, "hidden, secret"; and graphein means, "writing", or "study", respectively. It is the practice and study of techniques for secure communication in the presence of third parties. More generally, it is about constructing and analyzing protocols that block third parties; various aspects in information security such as data confidentiality, data integrity, authentication, and non-repudiation are central to modern cryptography.

Basically, the purpose of cryptography and steganography is to provide secret communication. However, steganography is not the same as cryptography. Cryptography hides the contents of a secret message from a malicious people, whereas steganography even conceals the existence of the message. Steganography must not be confused with cryptography, where we transform the message so as to make it meaning obscure to a malicious people who intercept it. In cryptography, the system is broken when the attacker can read the secret message. Breaking a steganographic system need the attacker to detect that steganography has been used and he is able to read the embedded message. In cryptography, the structure of a message is scrambled to make it meaningless and unintelligible unless the decryption key is available. It makes no attempt to disguise or hide the encoded message. Cryptography can also provide authentication for verifying the identity of someone or something.

In contrast, steganography does not alter the structure of the secret message, but hides it inside a cover-image so it cannot be seen. A message in cipher text, for instance, might arouse suspicion on the part of the recipient while an "invisible" message created with steganographic methods will not. It is possible to combine the techniques by encrypting message using cryptography and then hiding the encrypted message using steganography. The resulting stego-image can be transmitted without revealing that secret information is being exchanged. Furthermore, even if an attacker were to defeat the steganographic technique and detect the message from the stego-object, he would still require the cryptographic decoding key to decipher the encrypted message.

CHAPTER 2 DIGITAL IMAGE

A digital image is a numeric representation (normally binary) of a two-dimensional image. Images have a finite set of digital values, called picture elements or pixels. The digital image contains a fixed number of rows and columns of pixels. Pixels are the smallest individual element in an image, holding quantized values that represent the brightness of a given color at any specific point.

Typically, the pixels are stored in computer memory as a raster image or raster map, a two-dimensional array of small integers. These values are often transmitted or stored in a compressed form.

Raster images can be created by a variety of input devices and techniques, such as digital cameras, scanners, coordinate-measuring machines, seismographic profiling, airborne radar, and more. They can also be synthesized from arbitrary non-image data, such as mathematical functions or three-dimensional geometric models; the latter being a major sub-area of computer graphics. The field of digital image processing is the study of algorithms for their transformation.

2.1 Pixels and Bitmaps

Digital images are composed of pixels (short for picture elements). Each pixel represents the color (or gray level for black and white photos) at a single point in the image, so a pixel is like a tiny dot of a particular color. By measuring the color of an image at a large number of points, we can create a digital approximation of the image from which a copy of the original can be reconstructed. Pixels are a little like grain particles in a conventional photographic image, but arranged in a regular pattern of rows and columns and store information somewhat differently. A digital image is a rectangular array of pixels sometimes called a bitmap.

2.2 Types of Digital Images

For photographic purposes, there are two important types of digital images—color and black and white. Color images are made up of colored pixels while black and white images are made of pixels in different shades of gray.

A. Black and White Images

A black and white image is made up of pixels each of which holds a single number corresponding to the gray level of the image at a particular location. These gray levels span the full range from black to white in a series of very fine steps, normally 256 different grays. Since the eye can barely distinguish about 200 different gray levels, this is enough to give the illusion of a stepless tonal scale as illustrated below:

Figure 2. 1 0 to 256 gray levels

Assuming 256 gray levels, each black and white pixel can be stored in a single byte (8 bits) of memory.

B. Color Images

A color image is made up of pixels each of which holds three numbers corresponding to the red, green, and blue levels of the image at a particular location. Red, green, and blue (sometimes referred to as RGB) are the primary colors for mixing light—these so-called additive primary colors are different from the subtractive primary colors used for mixing paints (cyan, magenta, and yellow). Any color can be created by mixing the correct amounts of red, green, and blue light. Assuming 256 levels for each primary, each color pixel can be stored in three bytes (24 bits) of memory. This corresponds to roughly 16.7 million different possible colors. Note that for images of the same size, a black and white version will use three times less memory than a color version.

C. Binary or Bi-level Images

Binary images use only a single bit to represent each pixel. Since a bit can only exist in two states—on or off, every pixel in a binary image must be one of two colors, usually black or white. This inability to represent intermediate shades of gray is what limits their usefulness in dealing with photographic images.

2.3 Resolution

The more points at which we sample the image by measuring its color, the more detail we can capture. The density of pixels in an image is referred to as its *resolution*. The higher the resolution, the more information the image contains. If we keep the image size the same and increase the resolution, the image gets sharper and more detailed. Alternatively, with a higher resolution image, we can produce a larger image with the same amount of detail.

For example, the following images illustrate what happens as we reduce the resolution of an image while keeping its size the same—the pixels get larger and larger and there is less and less detail in the image:

Figure 2. 2 Original Image: Resolution (300,500)

Figure 2. 3 Half Size: Resolution (150,250)

Figure 2. 4 Quarter Size: Resolution (75,125)

Figure 2. 5 Resolution (37,62)

As we reduce the resolution of an image while keeping its pixels the same size—the image gets smaller and smaller while the amount of detail (per square inch) stays the same.

2.4 Compression

Image compression is minimizing the size in bytes of a graphics file without degrading the quality of the image to an unacceptable level. The reduction in file size allows more images to be stored in a given amount of disk or memory space. It also reduces the time required for images to be sent over the Internet or downloaded from Web pages.

There are several different ways in which image files can be compressed. For Internet use, the two most common compressed graphic image formats are the JPEG format and theGIF format. The JPEG method is more often used for photographs, while the GIF method is commonly used for line art and other images in which geometric shapes are relatively simple.

Other techniques for image compression include the use of fractals and wavelets. These methods have not gained widespread acceptance for use on the Internet as of this writing. However, both methods offer promise because they offer higher compression ratios than the JPEG or GIF methods for some types of images. Another new method that may in time replace the GIF format is the PNG format.

A text file or program can be compressed without the introduction of errors, but only up to a certain extent. This is called lossless *compression*. Beyond this point, errors are introduced. In text and program files, it is crucial that compression be lossless because a single error can seriously damage the meaning of a text file, or cause a program not to run. In image compression, a small loss in quality is usually not noticeable. There is no "critical point" up to which compression works perfectly, but beyond which it becomes impossible. When there is some tolerance for loss, the compression factor can be greater than it can when there is no loss tolerance. For this reason, graphic images can be compressed more than text files or programs.

CHAPTER 3 STEGANOGRAPHY

Since the rise of the Internet one of the most important factors of information technology and communication has been the security of information. Cryptography was created as a technique for securing the secrecy of communication and many different methods have been developed to encrypt and decrypt data in order to keep the message secret. Unfortunately it is sometimes not enough to keep the contents of a message secret, it may also be necessary to keep the existence of the message secret. The technique used to implement this, is called steganography.

Steganography is the art and science of invisible communication. This is accomplished through hiding information in other information, thus hiding the existence of the communicated information. The word steganography is derived from the Greek words "*stegos*" meaning "cover" and "*grafia*" meaning "writing" defining it as "covered writing". In image steganography the information is hidden exclusively in images.

Steganography differs from cryptography in the sense that where cryptography focuses on keeping the contents of a message secret, steganography focuses on keeping the existence of a message secret . Steganography and cryptography are both ways to protect information from unwanted parties but neither technology alone is perfect and can be compromised. Once the presence of hidden information is revealed or even suspected, the purpose of steganography is partly defeated . The strength of steganography can thus be amplified by combining it with cryptography.

3.1 History

The idea and practice of hiding information has a long history. In *Histories* the Greek historian Herodotus writes of a nobleman, Histaeus, who needed to communicate with his son-in-law in Greece. He shaved the head of one of his most trusted slaves and tattooed the message onto the slave's scalp. When the slave's hair grew back the slave was dispatched with the hidden message.

In the Second World War the Microdot technique was developed by the Germans. Information, especially photographs, was reduced in size until it was the size of a typed period. Extremely

difficult to detect, a normal cover message was sent over an insecure channel with one of the periods on the paper containing hidden information.

Today steganography is mostly used on computers with digital data being the carriers and networks being the high speed delivery channels.

Almost all digital file formats can be used for steganography, but the formats that are more suitable are those with a high degree of redundancy. Redundancy can be defined as the bits of an object that provide accuracy far greater than necessary for the object's use and display. The redundant bits of an object are those bits that can be altered without the alteration being detected easily. Image and audio files especially comply with this requirement, while research has also uncovered other file formats that can be used for information hiding.

3.2 Process flow of Image Steganography

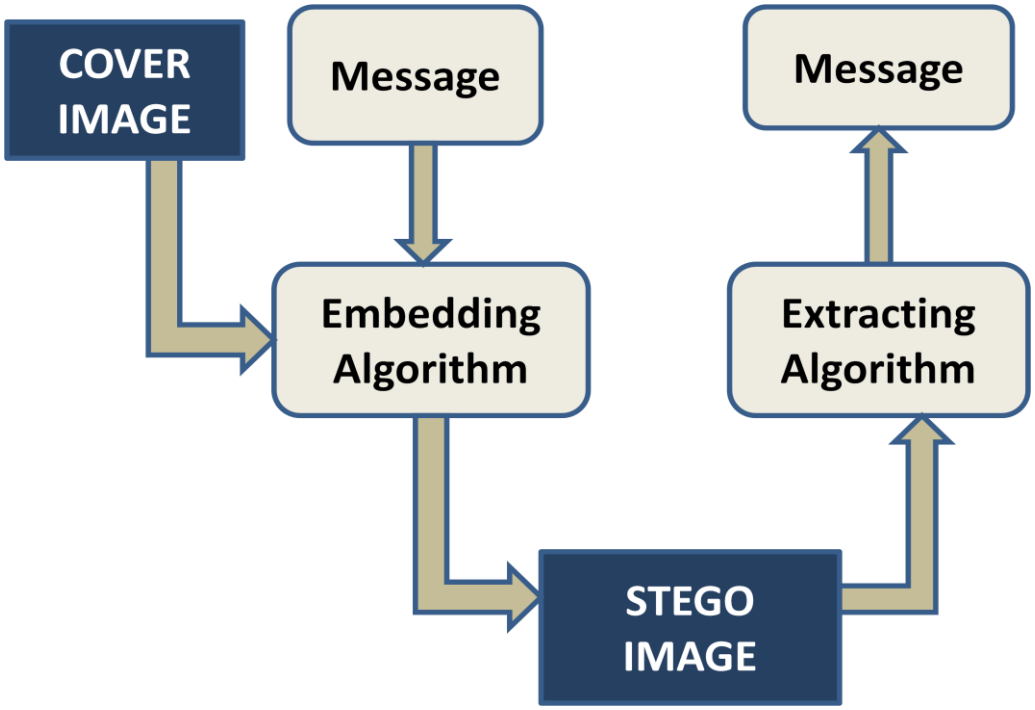


Figure 3. 1 Process Flow of Image Steganography

3.3 Types of Steganography

The four main categories of file formats that can be used for steganography:

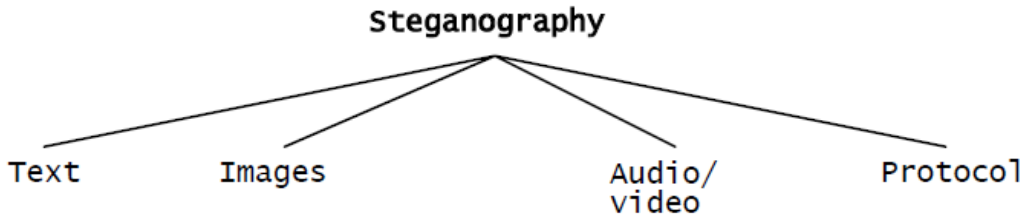


Figure 3. 2 Categories of Steganography

3.4 Image Steganography

Least Significant Bit Algorithm

Least significant bit (LSB) insertion is a common, simple approach to embedding information in a cover image. The least significant bit (in other words, the 8th bit) of some or all of the bytes inside an image is changed to a bit of the secret message. When using a 24-bit image, a bit of each of the red, green and blue colour components can be used, since they are each represented by a byte. In other words, one can store 3 bits in each pixel. An 800 × 600 pixel image, can thus store a total amount of 1,440,000 bits or 180,000 bytes of embedded data. For example a grid for 3 pixels of a 24-bit image can be as follows:

(00101101 00011100 11011100)

(10100110 11000100 00001100)

(11010010 10101101 01100011)

When the number 200, which binary representation is 11001000, is embedded into the least significant bits of this part of the image, the resulting grid is as follows:

(00101101 00011101 11011100)

(10100110 11000101 00001100)

(11010010 10101100 01100011)

Although the number was embedded into the first 8 bytes of the grid, only the 3 underlined bits needed to be changed according to the embedded message. On average, only half of the bits in an image will need to be modified to hide a secret message using the maximum cover size. Since there are 256 possible intensities of each primary colour, changing the LSB of a pixel results in small changes in the intensity of the colours. These changes cannot be perceived by the human eye - thus the message is successfully hidden. With a well-chosen image, one can even hide the message in the least as well as second to least significant bit and still not see the difference. In the above example, consecutive bytes of the image data – from the first byte to the end of the message – are used to embed the information. This approach is very easy to detect. A slightly more secure system is for the sender and receiver to share a secret key that specifies only certain pixels to be changed. Should an adversary suspect that LSB steganography has been used, he has no way of knowing which pixels to target without the secret key. In its simplest form, LSB makes use of BMP images, since they use lossless compression. Unfortunately to be able to hide a secret message inside a BMP file, one would require a very large cover image. Nowadays, BMP images of 800 × 600 pixels are not often used on the Internet and might arouse suspicion. For this reason, LSB steganography has also been developed for use with other image file formats.

- In LSB algorithm, the message bit is taken from the message byte and then that particular bit will be embedded inside the least significant bit of an image or video or audio file. This is done because
- The message embedded in the least significant bit of an image file will not draw the suspicion of the hacker as the minute difference that would be made in the pixel value of the image file will not be perceived by the normal naked human eye.

CHAPTER 4 CRYPTOGRAPHY

Cryptography is derived from Greek word kryptós means, "hidden, secret"; and graphein means, "writing", or "study", respectively. It is the practice and study of techniques for secure communication in the presence of third parties. More generally, it is about constructing and analyzing protocols that block third parties; various aspects in information security such as data confidentiality, data integrity, authentication, and non-repudiation are central to modern cryptography.

Data that can be read and understood without any special measures is called plaintext or cleartext. The method of disguising plaintext in such a way as to hide its substance is called encryption. Encrypting plaintext results in unreadable gibberish called ciphertext. You use encryption to ensure that information is hidden from anyone for whom it is not intended, even those who can see the encrypted data. The process of reverting ciphertext to its original plaintext is called decryption.

Cryptography is the art of protecting information by transforming it (encrypting it) into an unreadable format, called cipher text. Only those who possess a secret key can decipher (or decrypt) the message into plain text.

Cryptography is the science of using mathematics to encrypt and decrypt data. Cryptography enables you to store sensitive information or transmit it across insecure networks (like the Internet) so that it cannot be read by anyone except the intended recipient.

A cryptographic algorithm, or cipher, is a mathematical function used in the encryption and decryption process. A cryptographic algorithm works in combination with a key—a word, number, or phrase—to encrypt the plaintext. The same plaintext encrypts to different ciphertext with different keys. The security of encrypted data is entirely dependent on two things: the strength of the cryptographic algorithm and the secrecy of the key.

4.1 History

Before the modern era, cryptography was concerned solely with message confidentiality (i.e., encryption)—conversion of messages from a comprehensible form into an incomprehensible one and back again at the other end, rendering it unreadable by

interceptors or eavesdroppers without secret knowledge (namely the key needed for decryption of that message). Encryption was used to ensure secrecy in communications, such as those of spies, military leaders, and diplomats.

Suetonius tells us that Julius Caesar enciphered his dispatches by writing D for A, E for B and so on. When Augustus Caesar ascended the throne, he changed the imperial cipher system so that C was now written for A, D for B, and so on. In modern terminology, we would say that he changed the key from D to C. The Arabs generalized this idea to the monoalphabetic substitution, in which a keyword is used to permute the cipher alphabet.

In recent decades, the field has expanded beyond confidentiality concerns to include techniques for message integrity checking, sender/receiver identity authentication, digita signatures, interactive proofs and secure computation, among others

4.2 Types of Cryptography

- Private Key Cryptography
- Public Key Cryptography

KEY:

A key is a value that works with a cryptographic algorithm to produce a specific ciphertext. Keys are basically really, really, really big numbers. Key size is measured in bits; the number representing a 1024-bit key is darn huge. In public key cryptography, the bigger the key, the more secure the ciphertext.

Public key size and conventional cryptography's secret key size are totally unrelated. Bigger the key , more secure is the information.

a) Private Key Cryptography:

It is also known as symmetric cryptography and secret key cryptography. Symmetric-key cryptography refers to encryption methods in which both the sender and receiver share the same key. This was the only kind of encryption publicly known until June 1976.

The keys may be identical or there may be a simple transformation to go between the two keys. The keys, in practice, represent a shared secret between two or more parties that can be used to maintain a private information link. Moreover, symmetric encryption requires that a

secure channel be used to exchange the key, which seriously diminishes the usefulness of this kind of encryption system.

This requirement that both parties have access to the secret key is one of the main drawbacks of symmetric key encryption, in comparison to public-key encryption.

b) Public Key Cryptography:

The risk in symmetric system is that if either party loses the key or it is stolen, the system is broken. A more recent alternative is to use a combination of public and private keys. In this system, a public key is used together with a private key. It is also known as asymmetric cryptography.

A public key infrastructure (PKI) supports the distribution and identification of public encryption keys, enabling users and computers to both securely exchange data over networks such as the Internet and verify the identity of the other party. A public key infrastructure (PKI) supports the distribution and identification of public encryption keys, enabling users and computers to both securely exchange data over networks such as the Internet and verify the identity of the other party.

In public-key cryptosystems, the public key may be freely distributed, while its paired private key must remain secret. In a public-key encryption system, the public key is used for encryption, while the private or secret key is used for decryption.

Public key cryptography is an asymmetric scheme that uses a pair of keys for encryption: a public key, which encrypts data, and a corresponding private, or secret key for decryption. You publish your public key to the world while keeping your private key secret. Anyone with a copy of your public key can then encrypt information that only you can read. Even people you have never met.

It is computationally infeasible to deduce the private key from the public key. Anyone who has a public key can encrypt information but cannot decrypt it. Only the person who has the corresponding private key can decrypt the information.

The primary benefit of public key cryptography is that it allows people who have no preexisting security arrangement to exchange messages securely. The need for sender and receiver to share secret keys via some secure channel is eliminated; all communications involve only public keys, and no private key is ever transmitted or shared.

4.3 Public Key Cryptography

RSA Algorithm:

The acronym stands for Rivest, Shamir, and Adelman, the inventors of the technique. RSA is based on public-key cryptography. Public-key cryptography, also known as asymmetric cryptography, uses two different but mathematically linked keys one public and one private. RSA derives its security from the difficulty of factoring large integers that are the product of two large prime numbers.

To understand how the algorithm was designed, and why it works, we shall need several mathematical ingredients drawn from a branch of mathematics known as Number Theory, the study of whole numbers. Here are the ingredients we will draw from number theory:

- Modular arithmetic
- Fermat's "little" theorem
- The Euclidean Algorithm

Modular arithmetic:

The number a is equivalent (congruent) to the number b modulo n, or in shorthand notation $a \equiv b \pmod{n}$, if a differs from b by an exact multiple of n.

Examples:

- even numbers $\equiv 0 \pmod{2}$
- odd numbers $\equiv 1 \pmod{2}$
- $317 \equiv 2 \pmod{5}$
- $91 \equiv 1 \pmod{9}$
- $14 \equiv -1 \pmod{3}$

$a \equiv b \pmod{n}$

$a = nq + b$

Often a is given (and can be large), but we like to find b as small as possible, that is, between 0 and $n - 1$. In particular, if we need to make multiple modular calculations, we simplify them after each step, so that we won't need to multiply or add numbers bigger than $n - 1$. This

process of replacing a number with the remainder you get when you divide it by n is called reduction modulo n.

The RSA algorithm can be used for both key exchange and digital signatures. Although employed with numbers using hundreds of digits, the mathematics behind RSA is relatively straight-forward**.**

To create RSA public and private key pair, the following steps can be used:

I. Choose two prime numbers, p and q. From these numbers you can calculate the modulus, n = pq.

II. Select a third number, e, that is relatively prime to (i.e. it does not divide evenly into) the product $(p-1)(q-1)$, the number e is the public exponent.

III. Calculate an integer d from the quotient (ed-1)/((p-1)(q-1)). The number d is the private exponent.

IV. The public key is the number pair (n,e) . Although these values are publicly known, it is computationally infeasible to determine d from n and e if p and q are large enough.

V. To encrypt a message, M, with the public key, creates the cipher-text, C, using the equation: C= M^e Mod(n).

VI. The receiver then decrypts the cipher-text with the private key using the equation: M= C^d Mod(n).

Using Keys for encryption:

Assuming a sender "A" that wants to send a message to a receiver "B", the sender will take the following steps:

I. Obtains the recipient B's public key (e,n).

II. Represents the plaintext message as a positive integer M .

III. Computes the cipher-text C=M^e Mod(n).

IV. Send the cipher-text C to B.

RSA Encryption Algorithm:

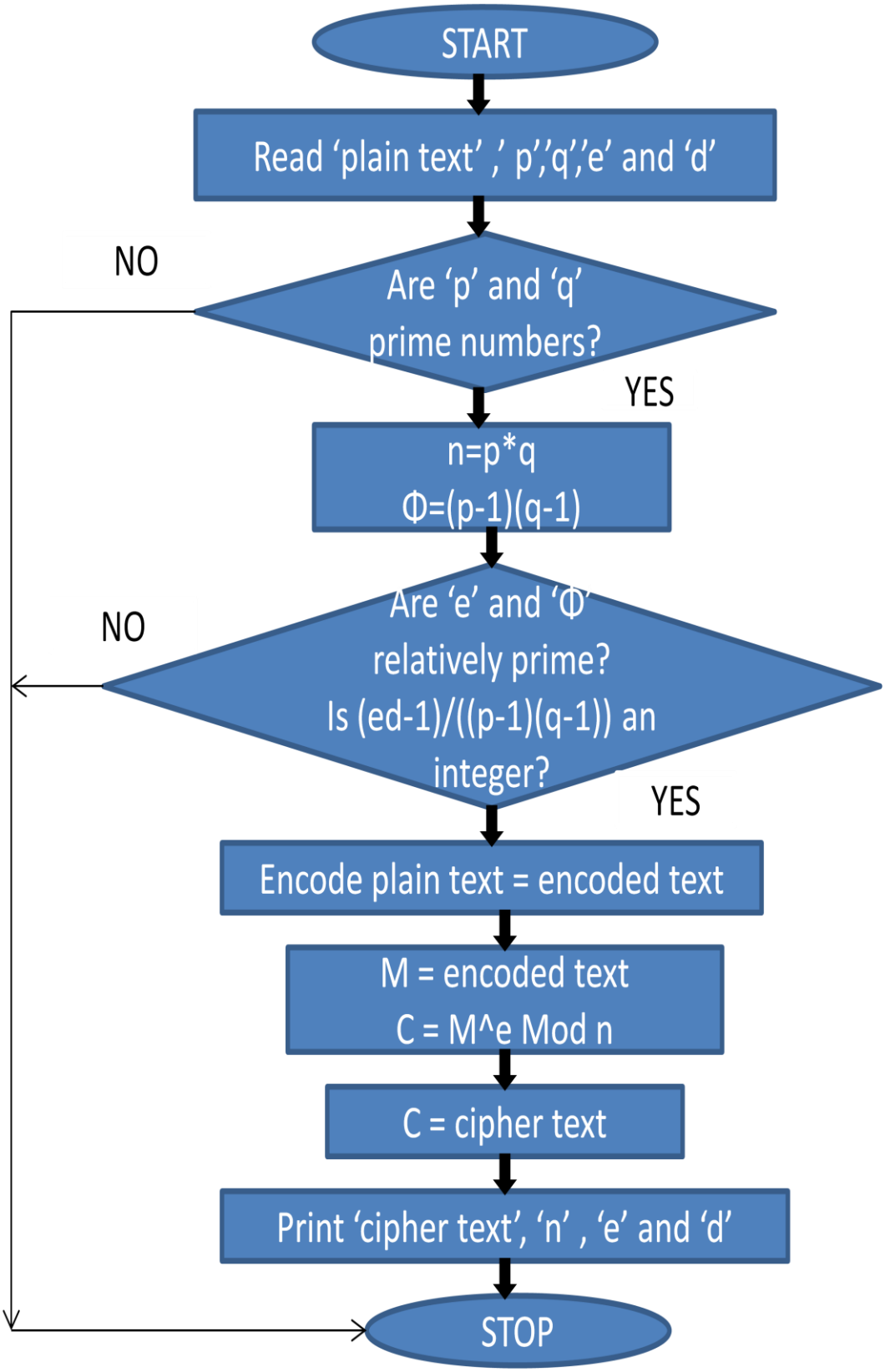


Figure 4. 1 RSA Encryption Algorithm

Using Keys for Decryption:

For the recipient “B” to receive the message sent by the sender “A”, the recipient will take the following steps:

I. Uses the private key (n, d) to compute M = C^e Mod(n).
II. Extracts the plaintext from the integer representative M.

This is actually the smallest possible value for the modulus n for which the RSA algorithm works.

RSA Decryption Algorithm:

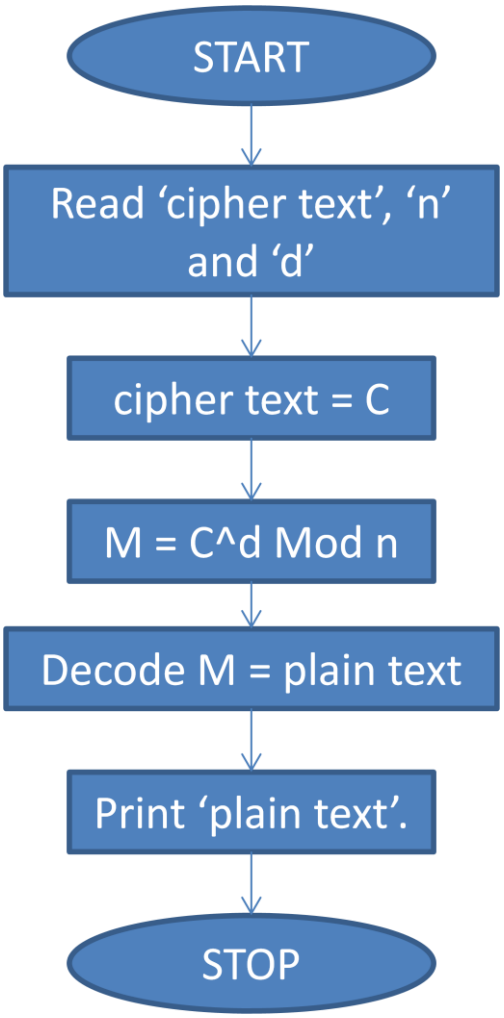


Figure 4. 2 RSA Decryption Algorithm

CHAPTER 5 IMPLEMENTATION AND RESULTS

5.1 Steganography vs. Cryptography

Basically, the purpose of cryptography and Steganography is to provide secret communication. However, Steganography is not the same as cryptography. Cryptography hides the contents of a secret message from a malicious people, whereas steganography even conceals the existence of the message. Steganography must not be confused with cryptography, where we transform the message so as to make it meaning obscure to a malicious people who intercept it. Therefore, the definition of breaking the system is different. In cryptography, the system is broken when the attacker can read the secret message. Breaking a steganographic system need the attacker to detect that steganography has been used and he is able to read the embedded message. In cryptography, the structure of a message is scrambled to make it meaningless and unintelligible unless the decryption key is available. It makes no attempt to disguise or hide the encoded message. Basically, cryptography offers the ability of transmitting information between persons in a way that prevents a third party from reading it. Cryptography can also provide authentication for verifying the identity of someone or something.

In contrast, steganography does not alter the structure of the secret message, but hides it inside a cover-image so it cannot be seen. A message in cipher text, for instance, might arouse suspicion on the part of the recipient while an “invisible” message created with steganographic methods will not. In other word, steganography prevents an unintended recipient from suspecting that the data exists. In addition, the security of classical steganography system relies on secrecy of the data encoding system. Once the encoding system is known, the steganography system is defeated. It is possible to combine the techniques by encrypting message using cryptography and then hiding the encrypted message using steganography. The resulting stego-image can be transmitted without revealing that secret information is being exchanged. Furthermore, even if an attacker were to defeat the steganographic technique and detect the message from the stego-object, he would still require the cryptographic decoding key to decipher the encrypted message.

Combining steganography and cryptography techniques is carried out in three steps:

- Encryption
- Steganography
- Decryption

5.2 Process Flow

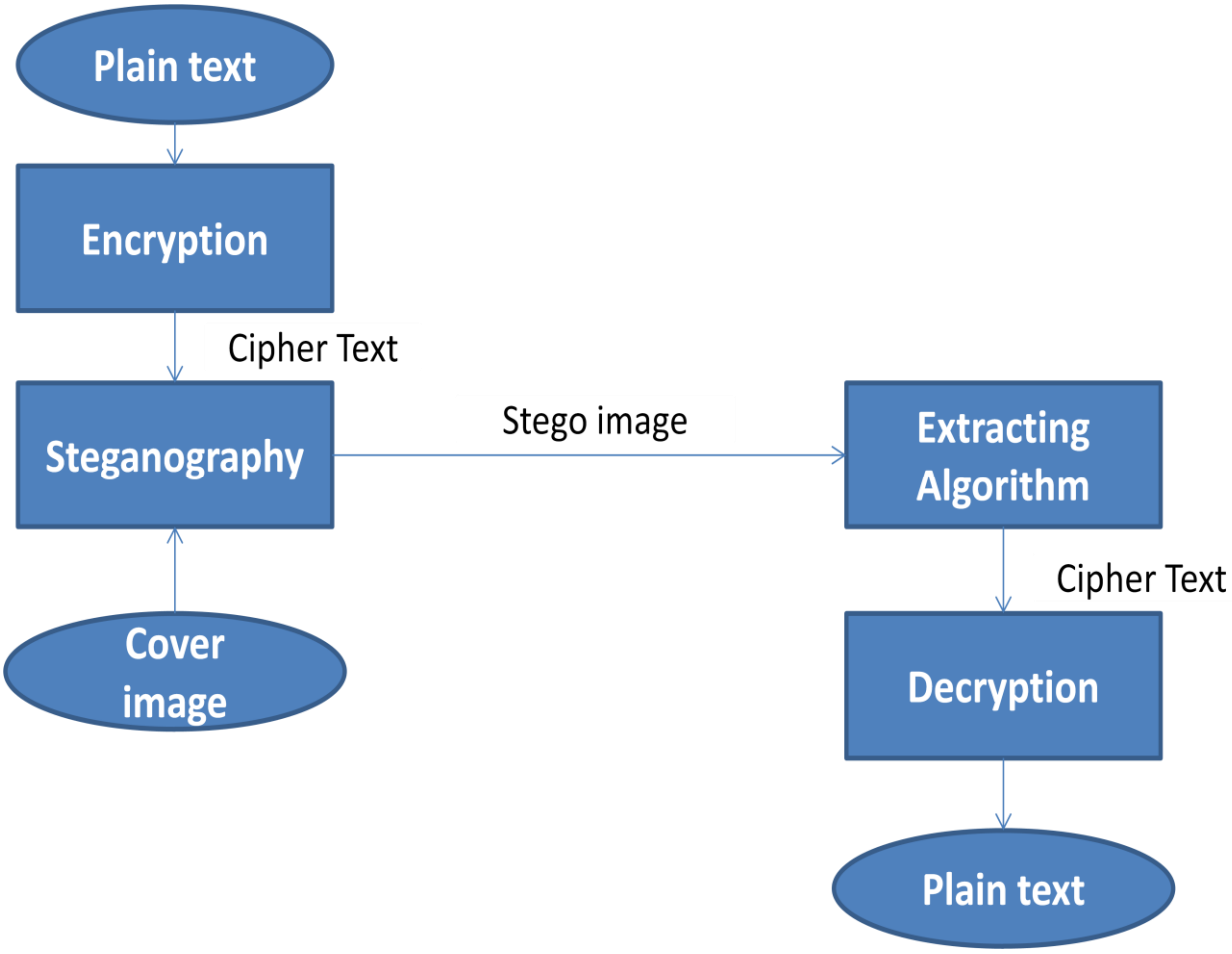


Figure 5. 1 Process Flow of combined Steganography and Cryptography

5.3 LSB Algorithm:

Embedding

Step 1: Read the Cover Image.

Step 2: Convert the image from RGB to gray.

Step 3: Take the secret message to be sent.

Step 4: Convert the message into Ascii Code.

Step 5: Convert it into Binary.

Step 6: Get the height and width of the image.

Step 7:

for i = 1 : height

for j = 1 : width

Extract LSB of each pixel till the size of message

If (LSB != message bit)

If (LSB==1)

Pixel value of Stego-Image = Pixel value of Original Image – 1

Else

Pixel value of Stego-Image = Pixel value of Original Image + 1

Else

Pixel value of Stego-Image = Pixel value of Original Image

Step 8: Save the Stego-image.

Extracting Algorithm:

Step 1: Read the Stego-Image.

Step 2: Get the height and width of the image

Step 3: for i = 1 : height

for j = 1 : width

Extract LSB of each pixel till the size of message

Step 4: Convert Binary (LSB values) to Decimal (AsciiCode)

Step 5: Convert AsciiCode into character

5.4 RSA Algorithm

RSA Encryption:

Step 1: Choose two prime numbers, p and q. From these numbers you can calculate the modulus, n = pq.

Step 2: Select a third number, e, that is relatively prime to (i.e. it does not divide evenly into) the product$(p-1)(q-1)$, the number e is the public exponent.

Step 3: Calculate an integer d from the quotient (ed-1)/((p-1)(q-1)). The number d is the private exponent.

Step 4: The public key is the number pair (n,e) . Although these values are publicly known, it is computationally infeasible to determine d from n and e if p and q are large enough.

Step 5: To encrypt a message, M, with the public key, creates the cipher-text, C, using the equation: C= M^e Mod(n).

RSA Decryption:

For the recipient "B" to receive the message sent by the sender "A", the recipient will take the following steps:

Step 1: Uses the private key (n, d) to compute M = C^e Mod(n).

Step 2: Extracts the plaintext from the integer representative M.

Combining:

- Converting the plain text into Cipher text
- Embed Cipher text into the cover image

5.5 Results

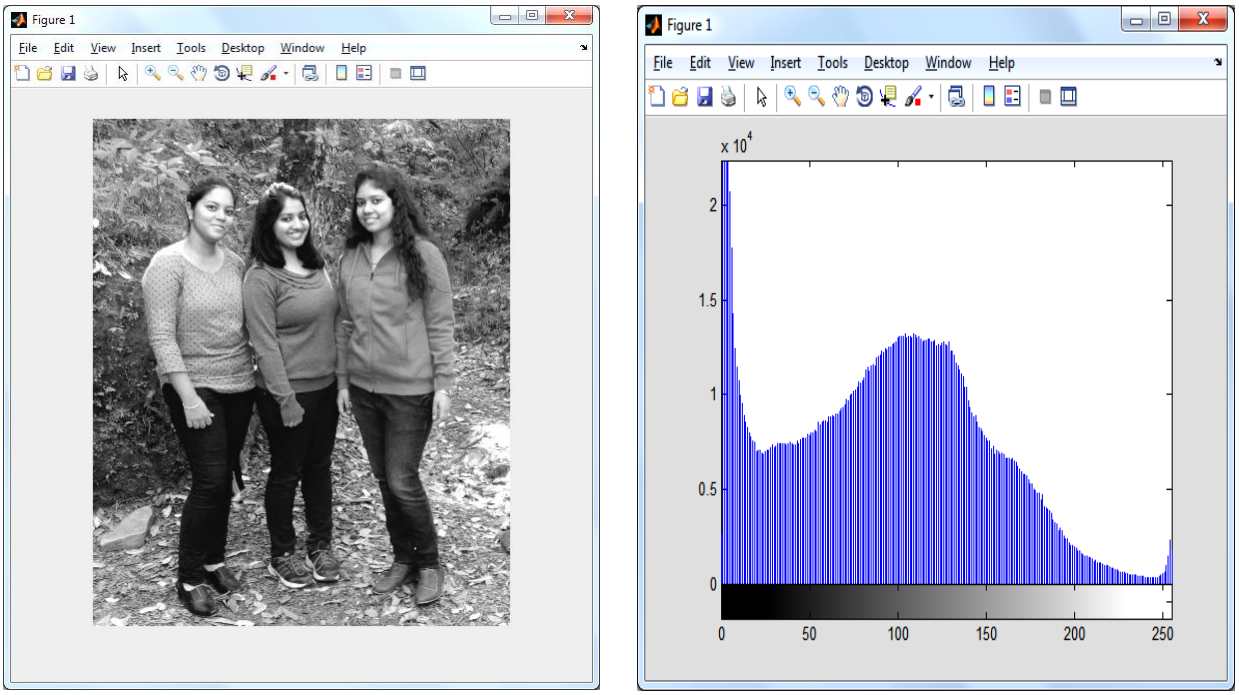


Figure 5. 2 Original Image and Histogram

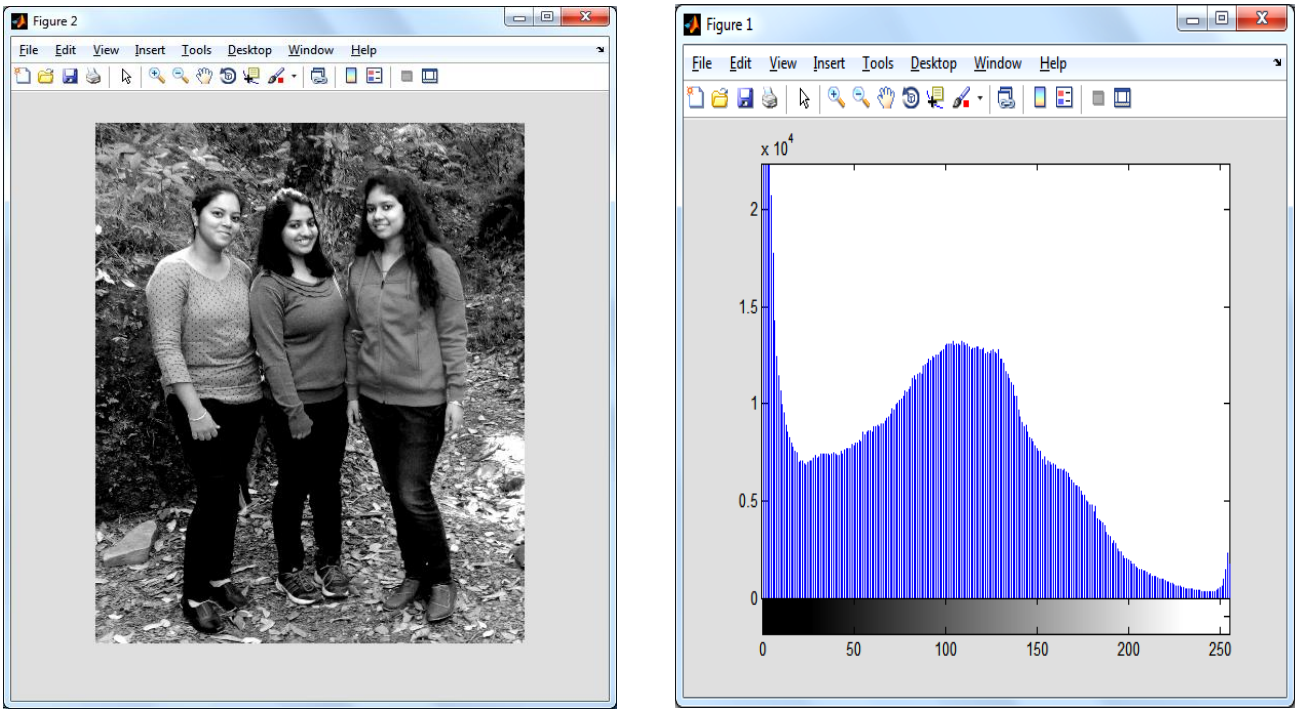


Figure 5. 3 Stego Image and Histogram

Message:

Your work is going to fill a large part of your life, and the only way to be truly satisfied is to do what you believe is great work. And the only way to do great work is to love what you do. If you haven't found it yet, keep looking. Don't settle. As with all matters of the heart, you'll know when you find it.

Cipher text

HLh-L-0ÑbgLÑgLUÑKK6K6-gº}6-LU1Lh-KÑUº96¬uºLK161Lsº-hK1b6ÑbUÑº¬ÑbL¬Lu61 LhsºKÑºvºÑbg-º6L-0%¶¬uºLK161L¬Lg-º6L-0ÑbLKLvºu61Lh¬L%cU1Lhu6vºAULh¬Ñ1º90º º}KLL0Ñg%ÌLAbºKº%¶bÑu6KK`6º-bLUuºuº6-91LhAKK0Luº1LhUÑ¬Ñ%

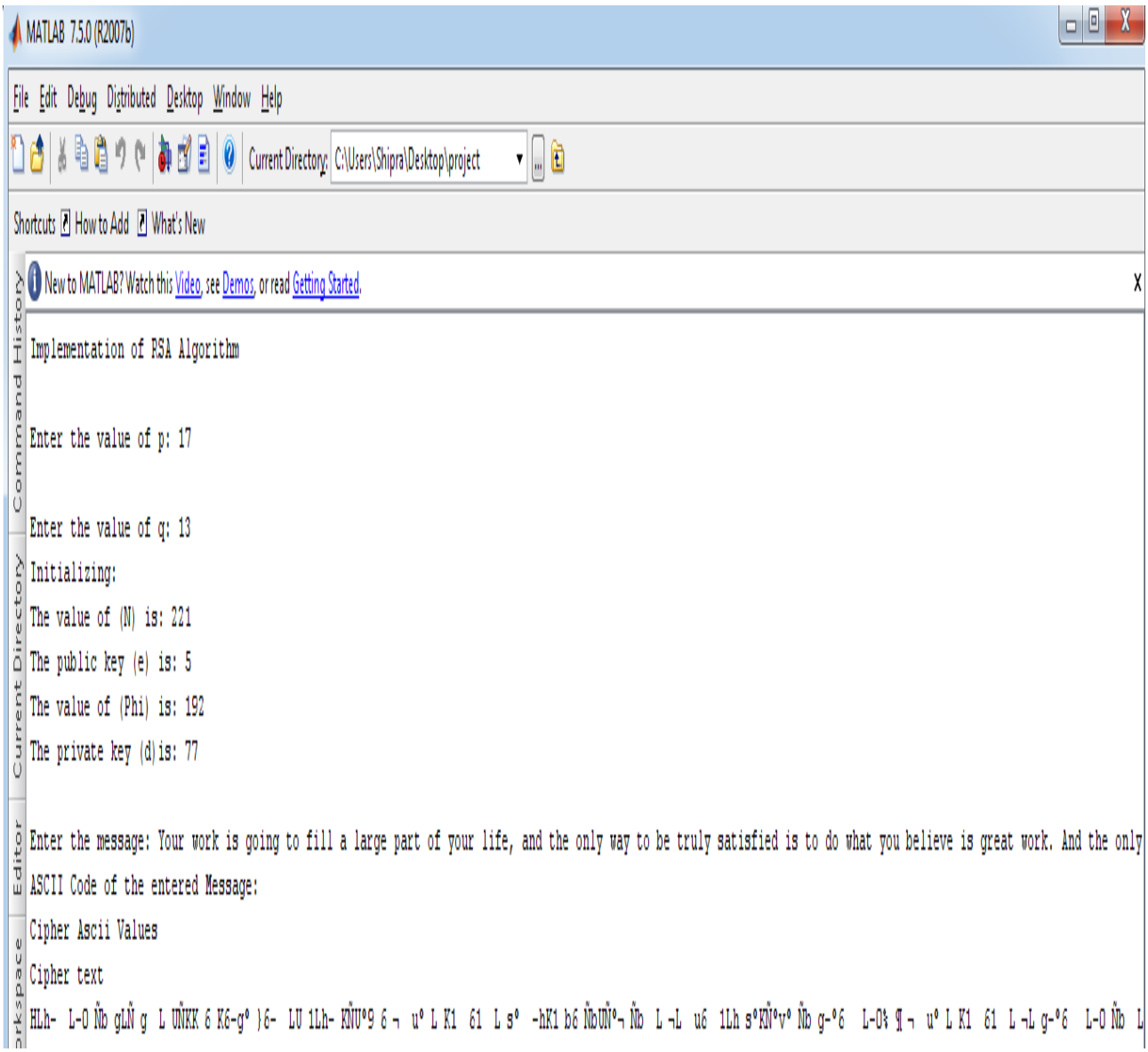


Figure 5. 4 Output: Cipher Text

5.6 Comparison:

Steps for Calculating Mean Square Error

1. Set up the data by using two images. One is stego image (I') and the other is original image (I).

2. Subtract stego image values(pixel pair values) from Original image values

3. Take the absolute value of each row. That is, if the difference is negative, remove the negative sign. If it is positive, leave it as is.

4. Add up the absolute values

5. Take the square of resultant

6. Divide by total number of rows and columns.ie., m x n .

In simple words MSE indicates average amount of modifications to the pixels.

For calculating the peak signal-to-noise ratio (PSNR)

PSNR = 10 * log 10 (256^2 / MSE)

In Image, PSNR value indicates the quality of stego image after modification.

Table 5.1

	LSB (%)	**Alternate LSB & SLSB(%)**	**Insertion using Quadratic Formula(%)**
MSE	**12.3**	**7.96**	**9.17**
PSNR	**37.2**	**39.15**	**36.23**

Table 5.2

	RSA + LSB(%)	**RSA + Alternate LSB & SLSB(%)**
MSE	**2.69**	**6.12**
PSNR	**43.8**	**40.29**

Limitations

LSB is very sensitive to any kind of filtering or manipulation of the stego-image. Scaling, rotation, cropping, addition of noise, or lossy compression to the stego-image will destroy the message. On the other hand, for the hiding capacity, the size of information to be hidden relatively depends to the size of the cover- image. The message size must be smaller than the image. A large capacity allows the use of the smaller cover-image for the message of fixed size, and thus decreases the bandwidth required to transmit the stego-image. Another weakness is an attacker can easily destruct the message by removing or zeroing the entire LSB plane with very little change in the perceptual quality of the modified stego-image. Therefore, if this method causes someone to suspect something hidden in the stego-image, then the method is not success.

CONCLUSION AND FUTURE SCOPE

Performance of LSB steganography for gray scale image and RSA Cryptography is analyzed. In addition to that, gray scale image is separated into layers and then data hiding is performed. The image quality after data embedding is very important for better performance of steganography methods. The image quality is evaluated by Mean Square Error (MSE) and Peak Signal to Noise ratio (PSNR) for gray-scale.This method encrypts the message, which improves the security. This scheme can be applied to other covers such as audio and video which is taken as the future work

Code

Encryption (Combined Steganography and Cryptography)

```
disp('Implementation of RSA Algorithm');
clear all; close all;
p = input('\nEnter the value of p: ');
q = input('\nEnter the value of q: ');

disp('Initializing:');
Pk=p*q;
Phi=(p-1)*(q-1);
%For Calculating the value of e

x=2;e=1;
while x > 1
    e=e+1;
    x=gcd(Phi,e);
end

%For Calculating the value of d
i=1;
r=1;
while r > 0
    k=(Phi*i)+1;
    r=rem(k,e);
    i=i+1;
end
d=k/e;

disp(['The value of (N) is: ' num2str(Pk)]);
disp(['The public key (e) is: ' num2str(e)]);
disp(['The value of (Phi) is: ' num2str(Phi)]);
disp(['The private key (d)is: ' num2str(d)]);

M = input('\nEnter the message: ','s');
x=length(M);
c=0;
for j= 1:x
    for i=0:255
        if strcmp(M(j),char(i))
            c(j)=i;
        end
    end
end
disp('ASCII Code of the entered Message:');

for j= 1:x
   cipher(j)= crypt(c(j),Pk,e);
end
```

```
disp('Cipher Ascii Values');

disp('Cipher text');
disp(char(cipher));

c1 = imread('girls.jpg');
coverimage = rgb2gray(c1);
imshow(coverimage); figure;

m = length(M) * 8;
fprintf('\nLength of message is : %d\n',m);

% Convert Message from decimal to binary
binaryString = transpose(dec2bin(cipher,8));

% Convert binaryString to Column Matrix
    binaryString = binaryString(:);

N = length(binaryString);

    %b is a Matrix having same number of bits as message
    b = zeros(N,1);

    %copy binaryString to b matrix
    for k = 1:N
        if(binaryString(k) == '1')
            b(k) = 1;
        else
            b(k) = 0;
        end
    end

    %Set Up for LSB

s = coverimage;  %s is original image

height = size(coverimage,1);
fprintf('\nheight = %d',height);

width = size(coverimage,2);
fprintf('\nwidth = %d\n',width);

    %LSB Algorithm
    %This goes to each byte, if the least significant bit is not the bit
of the message position, flip it, else do nothing
    k = 1;
    for i = 1 : height
        for j = 1 : width
            %extract LSB of each byte
             LSB = mod(double(coverimage(i,j)), 2);


            if (k>m || LSB == b(k))
                s(i,j) = coverimage(i,j);
                if(k<=m)
                                end
```

```
            else

                if(LSB == 1)
                    s(i,j) = coverimage(i,j) - 1;
                else
                    s(i,j) = coverimage(i,j) + 1;

                end
            end
            k = k + 1;
        end
    end

    sum = 0;
    for i = 1 : height
        for j = 1 : width
            diff = coverimage(i,j) - s(i,j);
            sum = sum + diff^2 ;
        end
    end

    disp(sum);
    disp(m);
    %Write image
    imwrite(s, 'hiddenmsgimage.bmp');
    imshow(s);
```

Decryption

```
%read into a matrix s
    s = imread('ab.bmp');

    height = size(s,1);
    width = size(s,2);

    m = 216;
b=0;
    %LSB Extraction
    %Go through each pixel data and save the least significant bit.
    k = 1;
    for i = 1 : height
        for j = 1 : width
            if (k <= m)
                b(k) = mod(double(s(i,j)),2);
                k = k + 1;
            end
        end
    end

    %Convert to string
    %Use a binary matrix multiply to do this
    binaryVector = b;
    binValues = [ 128 64 32 16 8 4 2 1 ];
    binaryVector = binaryVector';
```

```
    if mod(length(binaryVector),8) ~= 0
         error('Length of binary vector must be a multiple of 8.');
     end

    binMatrix = reshape(binaryVector,8,27);
    %disp(binMatrix);
   cipher = (binValues*binMatrix);

   for j= 1:27
    message(j)= crypt(cipher(j),Pk,d);
    % message(j)= mod(cipher(j)^d,Pk);
end
disp('Decrypted ASCII of Message:');
disp(message);

    %Print text
    disp(char(message));
```

Encoding (Steganography in alternate LSB and SLSB)

```
    clc;
    clear all;

    c1 = imread('dog.jpg');
    c = rgb2gray(c1);
    imshow(c); figure;

    message = 'Nitika Shipra';
    fprintf('\nMessage is : %s',message);

    %m is the length of the message in bits
     m = length(message) * 8;
     fprintf('\nLength of message is : %d\n',m);

    %Convert Message to ascii
    AsciiCode = uint8(message);
    fprintf('AsciiCode : ');
    disp(AsciiCode);

    %Convert Message from decimal to binary
    binaryString = transpose(dec2bin(AsciiCode,8));
    fprintf('\nbinaryString : \n');
    disp(binaryString);

    %Convert binaryString to Column Matrix
    binaryString = binaryString(:);
    fprintf('\nbinaryString : \n');
    disp(binaryString);

    N = length(binaryString);
```

```
    %b is a Matrix having same number of bits as message
    b = zeros(N,1);

    %copy binaryString to b matrix
    for k = 1:N
        if(binaryString(k) == '1')
            b(k) = 1;
        else
            b(k) = 0;
        end
    end
    fprintf('\nb : \n');
    disp(b);

    %Set Up for LSB

    s = c;  %s is original image

    height = size(c,1);
    fprintf('\nheight = %d',height);

    width = size(c,2);
    fprintf('\nwidth = %d\n',width);

    %LSB Algorithm
    %This goes to each byte, if the least significant bit is not the bit
of the message position, flip it, else do nothing
    k = 1;
    for i = 1 : height
        for j = 1 : width
            %extract LSB of each byte
            % LSB = mod(double(c(i,j)), 2);
             SLSB = mod(c(i,j), 4);
              if( SLSB== 0 || SLSB== 1)
                    a=0;
                else
                    a=1;
              end

              if (k>m || a == b(k) )
                s(i,j) = c(i,j);
              else if(SLSB == 0 || SLSB ==1)
                    s(i,j) = c(i,j) + 2;
                else
                            s(i,j) = c(i,j) -2;
                end
              end

            k = k + 1;
        end
    end

    sum = 0;
    for i = 1 : height
        for j = 1 : width
            diff = c(i,j) - s(i,j);
            sum = sum + diff^2 ;
```

```
    end
end

%Write image
imwrite(s, 'hiddenmsgimage.bmp');
imshow(s);
```

Extraction

```
%read into a matrix s
s = imread('hiddenmsgimage.bmp');

height = size(s,1);
width = size(s,2);

m = 104;

%LSB Extraction
%Go through each pixel data and save the least significant bit.
k = 1;
for i = 1 : height
    for j = 1 : width
        if (k <= m)
            b(k) = mod(double(s(i,j)),4);
                fprintf('\n\nmain\nb(%d)=%d',k,b(k));
            if( b(k)== 0 || b(k)== 1)
                b(k)=0;
                fprintf('\n 0 or 1\nb(%d)=%d',k,b(k));
            else
                b(k)=1;
                fprintf('\n 2 or 3\nb(%d)=%d',k,b(k));
            end
            k = k + 1;
        end
    end
end

%Convert to string
%Use a binary matrix multiply to do this
binaryVector = b;
binValues = [ 128 64 32 16 8 4 2 1 ];
binaryVector = binaryVector';

if mod(length(binaryVector),8) ~= 0
    error('Length of binary vector must be a multiple of 8.');
end

binMatrix = reshape(binaryVector,8,13);
%disp(binMatrix);

textString = (binValues*binMatrix);

%Print text
disp(char(textString));
```

REFERENCES:

1. Gandharba Swain and Saroj Kumar Lenka, "A Technique for Secret Communication Using a New Block Cipher with Dynamic Steganography", International Journal of Security and Its Applications Vol. 6, No. 2, April, 2012
2. Hemalatha S, U Dinesh Acharya, Renuka A, Priya R. Kamath,"A Secure And High Capacity Image Steganography Technique", An International Journal (SIPIJ) Vol.4, No.1, February 2013
3. Ramanpreet Kaur, Baljit Singh, Ishpreet Singh ,"A Comparative Study of Combination of Different Bit Positions In Image Steganography", Vol.2, Issue.5, Sep-Oct. 2012.
4. T. Morkel, J.H.P. Eloff , M.S. Olivier,"An Overview Of Image Steganography" Information and Computer Security Architecture (ICSA) Research Group Department of Computer Science University of Pretoria, 0002, Pretoria, South Africa.
5. Soumyendu Das, Subhendu Das, Bijoy Bandyopadhyay "Steganography and Steganalysis: Different Approaches".
6. Nentawe Y. Goshwe," Data Encryption and Decryption Using RSA Algorithm in a Network Environment", IJCSNS International Journal of Computer Science and Network Security, VOL.13 No.7, July 2013.
7. Dhawal Seth, L. Ramanathan, Abhishek Pandey, November 2010,"Security Enhancement: Combining Cryptography and Steganography", International Journal of Computer Applications (0975 –8887)Volume 9–No.11.
8. Rajan.S.Jamgekar, Geeta Shantanu Joshi," File Encryption and Decryption Using Secure RSA", International Journal of Emerging Science and Engineering (IJESE) ISSN: 2319–6378, Volume-1, Issue-4, February 2013.
9. http://www.datahide.com/BPCSe/applications-e.html
10. http://stegano.net/tutorial/steg-history.html
11. http://www.petitcolas.net/steganography/
12. http://www.jjtc.com/Steganography/
13. http://www.garykessler.net/library/steganography.html