ENHANCING SECURITY USING DIGITAL IMAGE PROCESSING

M.SAI LAXMI
A. SHRADHA
K. PRAGATHI

Department of Electronics and Communication Engineering

MAHATMA GANDHI INSTITUTE OF TECHNOLOGY
(Affiliated to Jawaharlal Nehru Technological University, Hyderabad, T.S.)

Chaitanya Bharathi P.O., Gandipet, Hyderabad – 500 075

2021

Enhancing Security Using Digital Image Processing

MINI PROJECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATIONENGINEERING

BY

M.SAI LAXMI-17VE1A0496
A. SHRADHA-18265A0413
K. PRAGATHI-18265A0417

Department of Electronics and Communication Engineering

MAHATMA GANDHI INSTITUTE OF TECHNOLOGY
(Affiliated to Jawaharlal Nehru Technological University, Hyderabad, T.S.)

Chaitanya Bharathi P.O., Gandipet, Hyderabad – 500075

2021

MAHATMA GANDHI INSTITUTE OF TECHNOLOGY

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad, T.S.)

Chaitanya Bharathi P.O., Gandipet, Hyderabad-500075

Department of Electronics and Communication Engineering

CERTIFICATE

Date: 20 February 2021

This is to certify that the Mini project work entitled **“Enhancing Security Using Digital Image Processing”** is a bonafide work carried out by

M.SAI LAXMI (17VE1A0496)
A. SHRADHA (18265A0413)
K. PRAGATHI (18265A0417)

in partial fulfillment of the requirements for the degree of **BACHELOR OF TECHNOLOGY** in **ELECTRONICS & COMMUNICATION ENGINEERING** by the Jawaharlal Nehru Technological University, Hyderabad during the academic year 2020-21.

The results embodied in this report have not been submitted to any other University or Institution for the award of any degree or diploma.

(Signature)
- - - - - - - - - - - - - - - - -
Mr. D.V.S. Nagendra Kumar
Asst.Professor
Faculty Advisor

(Signature)
- - - - - - - - - - - - - - -
Dr. S P Singh
Professor
Head of Dept.

ACKNOWLEDGEMENT

We express our deep sense of gratitude to our Guide Mr. D.V.S. Nagendra Kumar, Assistant Professor, for his invaluable guidance and encouragement in carrying out our Project.

We are highly indebted to our Faculty Co-ordinators, Dr.Ch. Raja, Dr.S. Srinivas Rao, Dr.Y. Praveen Kumar Reddy, in Electronics and Communication Engineering Department, who has given us all the necessary technical guidance in carrying out this Project.

We wish to express our sincere thanks to Dr. S.P. Singh, Head of the Department of Electronics and Communication Engineering, M.G.I.T, for encouraging us throughout the Project.

Finally, we thank all the people who have directly or indirectly help us through the course of our Project.

M.Sai Laxmi
A. Shradha
K. Pragathi

ABSTRACT

The problem of enhancing security of a secret image and secret message, which is to be sent over a network, by digitally processing it. We require that the secret image and secret message to be sent to the recipient in such a way that no one else suspects the existence of them. A cover image is used as a decoy in this technique in which the secret image as well as the secret message are embedded. Thus, the receiver obtains the secret image and the secret message from the cover image.

- As the text message is encrypted using AES algorithm and embedded in a part of the image the text message is difficult to find. More over since the secret image is broken down into parts and then sent to the receiver. This makes it difficult for the trespassers to get access to all the parts of the images at once.
- Thus, increasing the security to a much-needed higher level. This makes it becomes highly difficult for the intruder to detect and decode the document.

LIST OF CONTENTS

LIST OF FIGURES

CHAPTER -1

1.1. Introduction:

Lately, exponential growth of technology in every aspect of life is observed. Improvement of technology provides facilities to both users and hackers/intruders too. Advancement in technology that encourages hackers/intruders' activities result in lack of security to user's confidential data. The most common and popular techniques for data hiding that have been in use since long time are cryptography and steganography.

Enhancing security of a secret image and secret message, which is to be sent over a network, by digitally processing it. We require that the secret image and secret message to be sent to the recipient in such a way that no one else suspects the existence of them. A cover image is used as a decoy in this technique in which the secret image as well as the secret message are embedded. On the sender's side, the secret image is encrypted using AES Algorithm. In this encrypted secret image, the secret message is hidden using LSB Based Image Steganography.

Furthermore, the encrypted secret image with the secret text is hidden in the cover image, using LSB Based Image Steganography. The stego image thus obtained is split into 16 parts, indexed and sent to the receiver. On the receiver side, these sub images are fetched one by one and merged based on their index. The encrypted image is obtained from the merged image. Next, we extract the secret text from the LSBs of this encrypted image. Additionally, decryption is performed to extract the original secret image from the encrypted secret image. Thus, the receiver obtains the secret image and the secret message from the cover image.

1.2. Aim of the Project:

This project is developed for hiding information in any image file. The scope of the project is Implementation of steganography tools for hiding information, which includes any type of information file and image files and the path where the user wants to save the Image and extruded file.

1. To product security tool based on steganography techniques.
2. To explore techniques of hiding data using encryption module of this project
3. To extract techniques of getting secret data using decryption module.

1.3. Methodology:

One of the reasons that intruders can be successful is that most of the information they acquire from a system is in a form that they can read and comprehend. Intruders may reveal the information to others, modify it to misrepresent an individual or organization, or use it to launch an attack. One solution to this problem is through the use of steganography.

Steganography is a technique of hiding information in digital media. In contrast to cryptography, it is not to keep others from knowing the hidden information but it is to keep others from thinking that the information even exists. Steganography become more important as more people join the cyberspace revolution. Steganography is the art of concealing information in ways that prevents the detection of hidden messages. Steganography includes an array of secret communication methods that hide the message from being seen or discovered.

1.4. Significance of the Work:

In today’s world of growing technology security is of at most concern. With the increase in cyber-crime, providing only network security is not sufficient. Security provided to images like blue print of company projects, secret images of concern to the army or of company’s interest, using image steganography and stitching is beneficial.

Besides cryptography, steganography can be employed to secure information. In cryptography, the message or encrypted message is embedded in a digital host before passing it through the network, thus the existence of the message is unknown. Besides hiding data for confidentiality, this approach of information hiding can be extended to copyright protection for digital media: audio, video and images.

The growing possibilities of modern communications need the special means of security especially on computer network. The network security is becoming more important as the number of data being exchanged on the internet increases. Therefore, the confidentiality and data integrity are requiring to protect against unauthorized access and use. This has resulted in an explosive growth of the field of information hiding Information hiding is an emerging research area, which encompasses applications such as copyright protection for digital media, watermarking, fingerprinting, and steganography.

1.5. Organization of report:

- Chapter 1 contains the brief introduction of the field in which the project is built. Our project is one of the applications of digital image processing. Then the methodology how we approach the project to implement the process. Next the significance and the applications of the domain are discussed.

- Chapter 2 contains the introduction of the project where we discussed Traditional approach, Monoalphabetic cipher, MD5 Hashing, Data encryption standard, LSB substitution, Jsteg, Embedding process.

- Chapter 3 contains the history of the project. Mainly the history is based on Cryptanalysis and Stego analysis.

- Chapter 4 described the AES Algorithm and Inner working process of project. Coming to AES Algorithm we had discussed encryption algorithm and decryption algorithm and operations.

- Chapter 5 consists rationale, narrative and the problem statement.

- Chapter 6 contains introduction of Steganography in which we have discussed each and every step-in detail about steganography.

- Chapter 7 consists the details of process of project by flow of execution and and described in detail with each and every step being explored.

- Chapter 8 contains Analysis and Design of project.

CHAPTER -2

2.1. Traditional Approaches:

In this section, we discuss several encoding techniques that are often used in day-today functions to ensure data security. Currently, there are many methods which could hide data. All the methods may be applied at any time irrespective of the content available.

Each of these methods when used to encode and decode, data has its own constraints that need to be considered. There are certain requirements that must be satisfied. These requirements can be: the format of the input file, the size of the input file and the encryption key.

In the following sections, both cryptographic and steganographic methods that provide data confidentiality are described.

2.2. Monoalphabetic Ciphers:

Caesar cipher is the most popularly used substitution ciphering method. This was introduced by Suetonius in his biography [2] [3]. This involves a very simple substitution in which each alphabet will be replaced by third letter following it alphabetically. For example, if the plain text has an alphabet ‘A’ it will be replaced with ‘D’ and the same method is applied for all the letters in plaintext to produce a cipher text. All the letters in the alphabet are considered circularly (‘Z’ will be replaced by ‘C’), i.e., position of a letter in alphabet will be shifted by 3 positions. This shifting does not have to be 3, it can be any variable ‘k’. Though this technique has a key space of 26, it is found to be easy to break.

Definitive cryptography says that using a key will make the breaking process difficult because retaining the substitution will be time consuming as well usage keys were introduced into substitution. Here a key is considered and starting letters of the alphabet are substituted with the letters of the key. Remaining letters will be substituted by the letters that are not included in the key in alphabetical order. If a same letter appears again, it will simply be discarded.

2.3.MD5Hashing:

MD5 stands for message digest algorithm 5. This is a hashing algorithm that can be used as a digital signature mechanism. This is a widely used hashing algorithm whose hash value is 128-bits. This algorithm takes in a variable length input but gives a fixed length output of 128 bits. The given input is first divided into individual blocks of 512bits each. To make the blocks size divisible by 512, the last message block could be padded. While padding, first a single bit ‘1’isadded at the end and may be followed by many zeros until the blocks size can be divisible by 512. This algorithm uses four variables called state variables each of size 32-bits [5][9].

These state variables are initially stored with some default hexadecimal values. This algorithm also has four predefined functions that works on AND, OR, XOR and NOT operations. These functions use the state variables and message as input and convert the state variables from their original form to message digest. The generated digest will be stored in state variables. To get the final message digest, hexadecimal value of each state variable was taken as output. This is also called one-way hashing.

2.4. Data encryption standard (DES):

DES is an encryption standard where encryption is done in individual blocks called block cipher. The block size used here is 64 bits. The core idea behind this standard is Feistel network. This standard involves 16 identical stages in its process. Each block of 64 bits is divided into two blocks; left and right of 32-bits. The right part is given as input to a Feistel function. An XOR operation is applied between the output of Feistel function and the left part and the resultant is considered as the right part of the second stage. For the left part in the second stage, the right part from the previous stage is simply copied [11] [6].

The same procedure is contained for 16 iterations to get a final output of the 64-bit block. This is how each block of 64-bits is encrypted with DES.

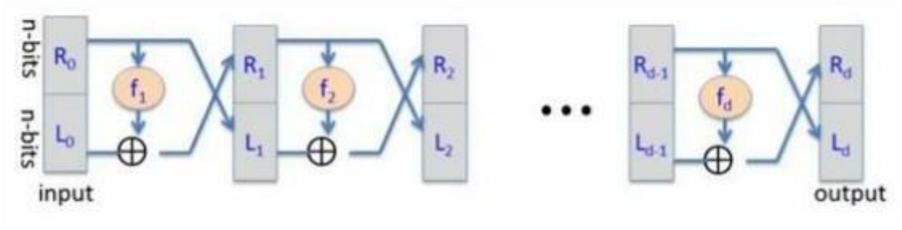


Fig 2.1: Encryption circuit block diagram

Decryption on the other hand, is just are verse process to encryption which can also be called as inversion where an XOR operation is applied between the right part and the output of the Feistel function. The result antis considered as the left part to the next stage. And the left part is just copied as the right part of the next stage. This process is carried out through all the 16stages.

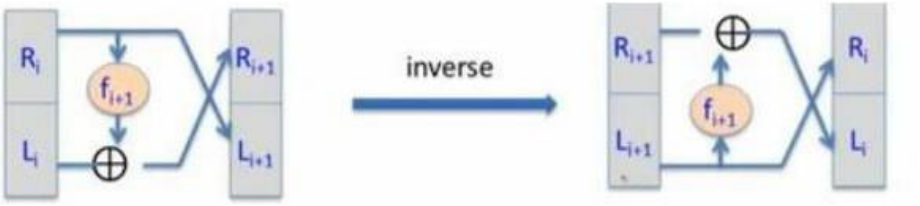


Fig 2.2: Inverse circuit of Encryption circuit

Feistel Function:

This function takes 32-bits of a block as input and submits to an expansion function. The task of the expansion function is to expand the input; therefore, this expansion function takes 32-bit information as input and gives 48-bit output. The Feistel function takes another input (key) which is of 48-bits length between the key value and 48-bit output from expansion function, an XOR operation is performed. From the resultant 48-bits, every 6 bits were given as input to an S-Box and a total of 8 S-boxes were used [11]. Each s-box 6 gives 4 bits of information as output; the output file is 32-bits in size.

2.5. Least Significant Bit Substitution (LSB):

LSB substitution is a popular technique to embed data on to digital images. We know that an image will be stored in the form of bytes. In this kind of encoding, by using the LSB of each byte,1-bit information can be stored in the image as secret message [8]. Accordingly,1-bitper bytecanbestoredin8-bitimageswhile3-bitscanbestoredin24-bitimages for every24-bits. Depending upon the color palette of a cover image, a secret message can be stored in two LSB's which cannot be identified by human visual system (HVS) [8] [9]. But the main drawback of this encoding method is that images after encoding can be intercepted easily i.e., information can be changed or image format can be changed.

2.6. Jsteg:

This was the first publicly available steganographic system for JPEG images. This encoding technique is similar to that of the LSB technique. This technique uses the concept of discrete cosine transformation (DCT) [11] [13]. The JPEG image format uses a discrete DCT to transform successive 8 × 8-pixel blocks of the image into 64 DCT coefficients each. Here, encoding is done by sequentially replacing the LSB of DCT coefficients with message's data. Andreas Westfield and Andreas Pittman noticed that steganographic systems that change least-significant bits sequentially cause distortions detectable by stainless [1] [8] [11]. The disadvantage with this system is, embedding step changes the LSB of colors in an image, that is, embedding uniformly distributed message bits reduces 7 the frequency difference between adjacent colors.

2.7. Embedding Process:

Interpreting technically, an image is stored in the form of an array of numbers on a computer that represents a collection of pixels, representing light intensity at various points of the image. Typically, a pixel could carry 8-bits or 24-bits of data depending upon the image quality. An 8-bit image is smaller in size which can be dependable, but one disadvantage is that, it exhibits only 256 colors. This may affect adversely while encoding.

Hence, while handling an 8-bit image, gray scale color palette is used. Use of 24-bit images increases efficiency because it can exhibit many colors(morethan16million) [10][12]. A file with bigger size (usually in Mega Bytes) can make it more suspicious when transmitted on the internet.

CHAPTER -3

3. Literature Review

3.1. Cryptanalysis:

Apart from encoding techniques, there are some universal tricks that are capable of breaking ciphertext into plaintext without any clue of keys or what the message has. This is process of decoding a ciphertext is called cryptanalysis, that is a counter technique to cryptographic methods. The base idea behind cryptanalysis is repetition of letters. Every language will have many words to exchange information [3].

For example, English alphabet has only 26 letters but combination of those letters can give many words: enough to communicate. This idea gave birth to ‘frequency analysis’ technique in cryptanalysis. 8 In the process of cryptanalysis, an analysis will check for the distribution of frequency of the letters in the cipher text and compare it with the distribution of the frequency of normal alphabet.

Looking at the comparison results, the analyst will substitute appropriate letters to get plaintext. The letter with more frequency in English is ‘e’. There are many tools that work on character substitution techniques, CRANK is the popular attacking tool for this technique [3][4].

3.2. Stego Analysis:

Attacking techniques for steganography were also developed. When compared to cryptanalysis, stainless is a more challenging task for the analyst, because ciphertext can easily be identified by looking at it. Whereas a message hidden in an image cannot be determined or even look suspicious to the eye of the analyst. There are three attacks possible which are considered as stainless: Visual attack, Structural attack and Statistical attack [1] [3].

The simplest form of stainless is visual attack, in which the analyst’s attack is by observing subjected file with naked eyes. The first rule of good steganography is to keep the stego-image unchanged. Despite of that, removing some parts of the images that were not altered and just focus on the altered parts make the visual attack a successful one. Therefore, the main idea of this attack is to identify which parts of the stego-image are to be considered.

The study of image says that, in an image, the bits with value 1 are almost equal to bits with value 0. That is there are approximately equal number of 0's and 1's in an image. Embedding of some plain data will disprove this fundamental rule; but if the embedded data is encrypted, then the fundamental rule will remain the same.

Structural attacks mainly focus on the high-level properties of a particular method or algorithm. However, when we remove parts of the image that are not altered as a result of embedding a message, and instead concentrate on the likely area so embedding in isolation, it is usually possible to observe signs of manipulation. The 'Hide & Seek' steganographic algorithm can be applied only to images whose size is 320 × 480 pixels, which is an example of this attack. In this case, an analyst can consider all the images satisfying those size specifications as a suspicious image to perform his attack.

Statistical attacks are mathematical approaches that involve some statistical approach. In a data set the presence of some peculiar data which is random, can be identified using Statistics. With the help of this strategy the analyst could break the stego image in to two sets, one is image data and the other will be information data. All the information regarding the image that can be seen, which is fully related to pixel values and the colors of those pixels are included in image data. Information about the hidden message comes under information data.

CHAPTER-4

4.1. AES Algorithm:

The more popular and widely adopted symmetric encryption algorithm likely to be encountered nowadays is the Advanced Encryption Standard (AES). It is found at least six time faster than triple DES.

A replacement for DES was needed as its key size was too small. With increasing computing power, it was considered vulnerable against exhaustive key search attack. Triple DES was designed to overcome this drawback but it was found slow.

The features of AES are as follows −

- Symmetric key symmetric block cipher
- 128-bit data, 128/192/256-bitkeys
- Stronger and faster than Triple-DES
- Provide full specification and design details
- Software implementable in C and Java

AES algorithm is of three types i.e., AES-128, AES-192 and AES-256. This classification is done on the bases of the key used in the algorithm for encryption and decryption process. The numbers represent the size of key in bits. This key size determines the security level as the size of key increases the level of security increases.

4.1.1. Encryption Algorithm:

The implementation of the AES-128 encryption and decryption algorithm with the help of MATLAB software is Fig.6. Flowchart of AES Encryption algorithm done. In which the input is an image and the key in hexadecimal format and the output is the same as that of input image. For encryption process first, dividing image and making it 4*4-byte state i.e., matrix format. Calculate the number of rounds based on the key Size and expand the key using our key schedule. And there are (n-1) rounds performed which are substitute byte, shift rows, mix columns and add round key. The final round “n” does not consist of mix column in the iteration. Figure 6 shows the flow of algorithm.

4.1.2. Decryption Algorithm:

The AES decryption process is the revers process that of the encryption process. The above figure shows flow of the AES decryption algorithm. Which consist of cipher text as the input, the key is same for decryption process which for encryption. In case of decryption the inverse substitute byte, inverse shift rows and the inverse mix columns are to be implemented. While the add round key remains the same.

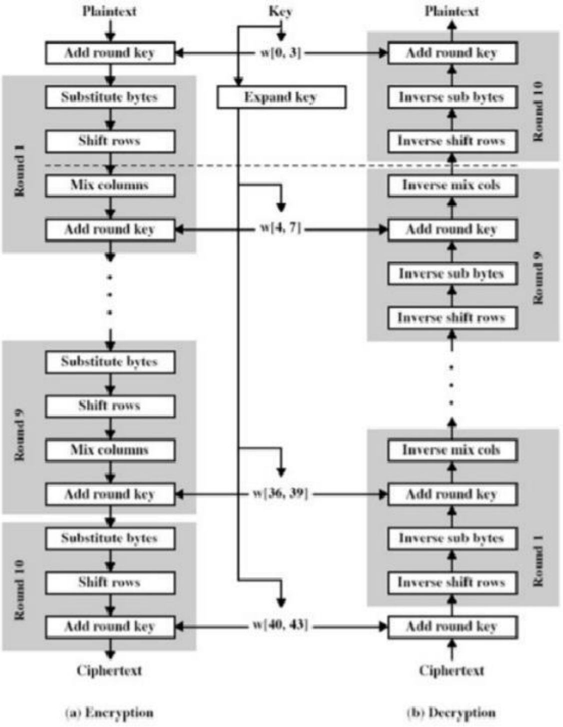


Fig 4.1: Overall structure of the AES algorithm

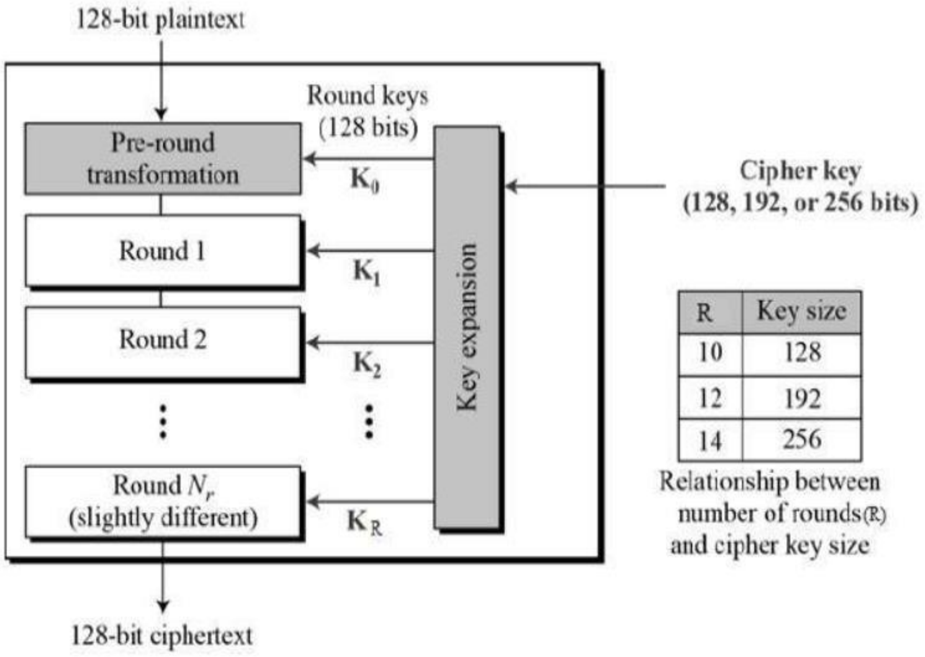
4.1.3. Operation of AES:

AES is an iterative rather than Feistel cipher. It is based on ‘substitution–permutation network’. It comprises of a series of linked operations, some of which involve replacing inputs by specific outputs (substitutions) and others involve shuffling bits around (permutations).

Interestingly, AES performs all its computations on bytes rather than bits. Hence, AES treats the 128 bits of a plaintext block as 16 bytes. These 16 bytes are arranged in four columns and four rows for processing as a matrix –

Unlike DES, the number of rounds in AES is variable and depends on the length of the key. AES uses 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. Each of these rounds uses a different 128-bit round key, which is calculated from the original AES key.

The schematic of AES structure is given in the following illustration –



R	Key size
10	128
12	192
14	256

Relationship between number of rounds(R) and cipher key size

Fig 4.2: schematic of AES structure

4.2. Inner Workings of a Round:

The algorithm begins with an Add round key stage followed by 9 rounds of four stages and a tenth round of three stages. This applies for both encryption and decryption with the exception that each stage of a round the decryption algorithm is the inverse of its counterpart in the encryption algorithm.

The four stages are as follows:

1. Substitute bytes

2. Shift rows

3. Mix Columns

4. Add Round Key

The tenth round simply leaves out the Mix Columns stage. The first nine rounds of the decryption algorithm consist of the following:

1. Inverse Shift rows
2. Inverse Substitute bytes
3. Inverse Add Round Key
4. Inverse Mix Columns

Again, the tenth round simply leaves out the Inverse Mix Columns stage. Each of these stages will now be considered in more detail.

4.2.1. Encryption Process:

Here, we restrict to description of a typical round of AES encryption. Each round comprises of four sub-processes. The first-round process is depicted below –

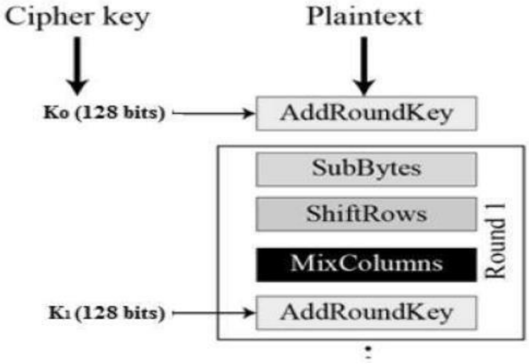


Fig 4.3: Encryption Process

Byte Substitution (Sub Bytes):

The16 input bytes are substituted by looking up a fixed table(S-box) given in design. The result is in a matrix of four rows and four columns.

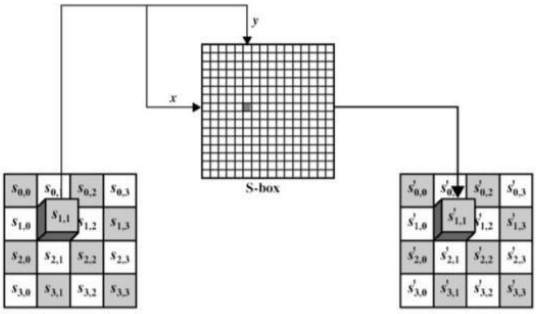

y
x
S-box
$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,0}$ $s_{1,1}$ $s_{1,2}$ $s_{1,3}$
$s_{2,0}$ $s_{2,1}$ $s_{2,2}$ $s_{2,3}$
$s_{3,0}$ $s_{3,1}$ $s_{3,2}$ $s_{3,3}$
$s'_{0,0}$ $s'_{0,1}$ $s'_{0,2}$ $s'_{0,3}$
$s'_{1,0}$ $s'_{1,1}$ $s'_{1,2}$ $s'_{1,3}$
$s'_{2,0}$ $s'_{2,1}$ $s'_{2,2}$ $s'_{2,3}$
$s'_{3,0}$ $s'_{3,1}$ $s'_{3,2}$ $s'_{3,3}$

Fig4.4: Substitute Bytes Stage of the AES algorithm.

Shift rows:

Each of the four rows of the matrix is shifted to the left. Any entries that ‘fall off’ are re-inserted on the right side of row. Shift is carried out as follows −

- ➢ First row is not shifted.
- ➢ Second row is shifted one (byte) position to the left.
- ➢ Third row is shifted two positions to the left.
- ➢ Fourth row is shifted three positions to the left.

The result is a new matrix consisting of the same 16 bytes but shifted with respect to each other.

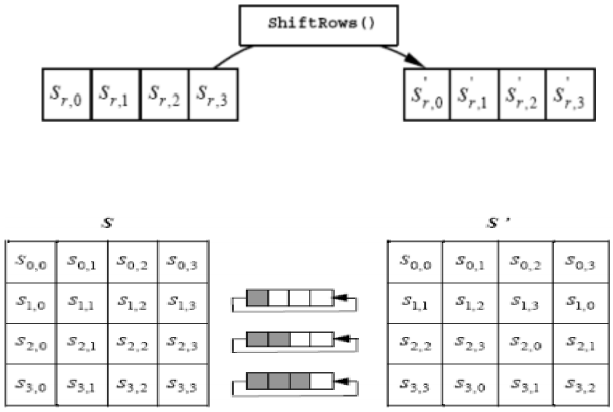

ShiftRows()
$S_{r,0}$ $S_{r,1}$ $S_{r,2}$ $S_{r,3}$
$S'_{r,0}$ $S'_{r,1}$ $S'_{r,2}$ $S'_{r,3}$
s
s'
$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,0}$ $s_{1,1}$ $s_{1,2}$ $s_{1,3}$
$s_{2,0}$ $s_{2,1}$ $s_{2,2}$ $s_{2,3}$
$s_{3,0}$ $s_{3,1}$ $s_{3,2}$ $s_{3,3}$
$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,1}$ $s_{1,2}$ $s_{1,3}$ $s_{1,0}$
$s_{2,2}$ $s_{2,3}$ $s_{2,0}$ $s_{2,1}$
$s_{3,3}$ $s_{3,0}$ $s_{3,1}$ $s_{3,2}$

Fig4.5: Cyclic shift row operation

Mix Columns:

Each column off our bytes is now transformed using a special mathematical function. This function takes as input the four bytes of one column and outputs four completely new bytes, which replace the original column. The result is another new matrix consisting of 16 new bytes. It should be noted that this step is not performed in the last round.

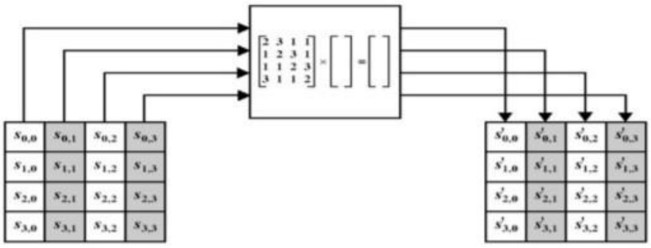


Figure.4.6 :Mix columns operation

Add round key:

The 16bytes of the matrix are now considered as128bits and XORed tothe128 bits of the round key. If this is the last round then the output is the ciphertext. Otherwise, the resulting 128 bits are interpreted as 16 bytes and we begin another similar round.

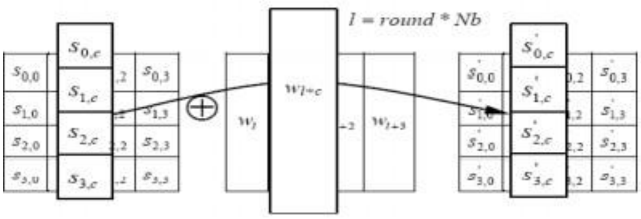


Figure 4.7: Add round key operation

4.2.2. Decryption Process:

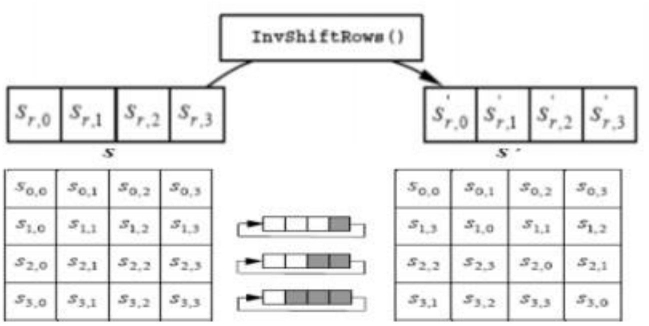
The process of decryption of an AES cipher text is similar to the encryption process in the reverse order. Each round consists of the four processes conducted in the reverse order,

1. Inverse shift row
2. Inverse substitute byte
3. Add round key
4. Inverse mix columns

Since sub-processes in each round are in reverse manner, unlike for a Feistel Cipher, the encryption and decryption algorithms need to be separately implemented, although they are very closely related.

Inverse Shift Rows:

Inverse Shift Rows is the inverse of the Shift Rows transformation. The bytes in the last three rows of the State are cyclically shifted over different numbers of bytes. The first row, r= 0, is not shifted. The bottom three rows are cyclically shifted by Nb-shift (r, Nb) bytes, where the shift value shift (r, Nb) depends on the row number.


InvShiftRows()
$s_{r,0}$ $s_{r,1}$ $s_{r,2}$ $s_{r,3}$
s
$s'_{r,0}$ $s'_{r,1}$ $s'_{r,2}$ $s'_{r,3}$
s'

$s_{0,0}$	$s_{0,1}$	$s_{0,2}$	$s_{0,3}$
$s_{1,0}$	$s_{1,1}$	$s_{1,2}$	$s_{1,3}$
$s_{2,0}$	$s_{2,1}$	$s_{2,2}$	$s_{2,3}$
$s_{3,0}$	$s_{3,1}$	$s_{3,2}$	$s_{3,3}$

$s_{0,0}$	$s_{0,1}$	$s_{0,2}$	$s_{0,3}$
$s_{1,3}$	$s_{1,0}$	$s_{1,1}$	$s_{1,2}$
$s_{2,2}$	$s_{2,3}$	$s_{2,0}$	$s_{2,1}$
$s_{3,1}$	$s_{3,2}$	$s_{3,3}$	$s_{3,0}$

Figure.4.8: Inverse Shift row operation

Inverse substitute byte transformation:

Inverse Substitute Bytes is the inverse of the byte substitution transformation, in which the inverse S-box is applied to each byte of the State. It is reverse process of Substitute byte transform. This is obtained by applying the inverse of the affine transformation followed by taking the multiplicative inverse in GF (2^8). There is an inverses-box table for substitute the value.

Inverse mix columns transformation:

Inverse Mix Columns is the inverse of the Mix Columns transformation. Inverse Mix Columns operates on the State column-by-column, treating each column as a four-term polynomial. The columns are considered as polynomials over GF (2^8) and multiplied modulo x^ 4 + 1 with a fixed polynomial (x), given by

a-1(x) = {0b}x^3 + {0d}x^ 2 + {09}x + {0e}

CHAPTER-5

5.1. Rationale

Initially, there were not many tools that could block the cryptanalysis or stego analysis attempts at various levels to secure/protect the hidden data in text or carrier media respectively. There were many situations this tool was needed. For, example attacks on military secrets and plans were occurred many times in many countries. Data hiding is a very important viewpoint of any user who transmits confidential information through networks.

Sometimes trade secrets may have to be shared among different industries. In the same way, there can be many situations that require data hiding.

Though there were many methods introduced for data hiding, still there has been a challenge. Many of them were designed in a way that follows the approach which shows a small change to previously introduced approaches. Each of them has their own disadvantages. Few of which are discussed in this document.

Digital images are being used as most common carriers to hide secret messages because the human visual system (HVS) has limitations. That is, the human visual system has no function like abstracting the illusion effects on what it looks [1] [12]. Steganography took advantage of using digital images so that small visual changes to an image cannot be suspected. Essentially any media such as plaintext, ciphertext, can be hidden in a digital image that can be converted into a bitstream.

The new tool proposer here is a combination of both cryptography and steganography. This tool uses a digital image as a carrier to take advantage over HVS. This tool would try blocking an analyst through multiple levels up to maximum extent. The mechanism of this new tool is discussed below along with some screen shots of the tool while running it.

5.2. Problem Statement:

Currently many cryptography and steganography techniques have come into existence. Encoding of plaintext is achieved using DES, AES, Triple DES, RSA and many other algorithms. Any individual can use his/her one’s own approach as encryption method. Many algorithms such as Jsteg, JP Hide and JP Seek, Outguess, F3, F4 and F5 were invented for the purpose of embedding images. These algorithms follow a certain principle to embed and retrieve hidden contents.

All the existing approaches have their own disadvantages as they can easily be compromised using stego analysis. It means that one way or another, an intruder can figure out the existence of hidden data which results in him/her compromise of sensitive data. Currently, no integrated cryptography and steganography approach in one application exists for image-based information security. There are encryption and embedding approaches present that work with plaintext only.

5.2.1. Motivation:

As described above all the available techniques used in early tools are old and follow some specified process with some improvements to previously proposed techniques. This makes the intruders work easy. The intruder may try a counter attack by making some changes to counter existing techniques. None of the existing techniques offers protection through multiple levels. That is one of the reasons why an intruder is able to view/obtain hidden data with just one or two attacks.

5.2.2. Scope:

The primary idea behind developing this project is to protect confidential data from an intruder's counter-attacks and to block the intruder through various levels in his/her attacks. A new tool has been developed with a combination of cryptographic encryption and steganographic encryption for its implementation. The developed steganographic tool has a sender's segment that can take a message, a password and a cover image as input and give a stego-image as output that has message embedded in it. On the other hand, it also has a receiver's segment where the receiver inputs the stego-image and the same password is used by the sender as input to get the sender's message as output. The project is tested with various inputs and made sure that the generated stego-image has no noise are data loss.

CHAPTER-6

6.1. Steganography:

Data hiding is of importance in many applications. For hobbyists, secretive data transmission, for privacy of users etc. the basic methods are: Steganography and Cryptography.

Steganography is a simple security method. Generally, there are three different methods used for hiding information: steganography, cryptography, watermarking.

In cryptography, the information to be hidden is encoded using certain techniques. this information is generally understood to be coded as the data appears nonsensical.

Steganography is hiding information; this generally cannot be identified because the coded information doesn’t appear to be abnormal i.e., its presence is undetectable by sight.

Detection of steganography is called Stego analysis. Steganography is of different types:

- Text steganography
- Image steganography
- Audio steganography
- Video steganography

In all of these methods, the basic principle of steganography is that a secret message is to be embedded in another cover object which may not be of any significance in such a way that the encrypted data would finally display only the cover data. So, it cannot be detected easily to be containing hidden information unless proper decryption is used.

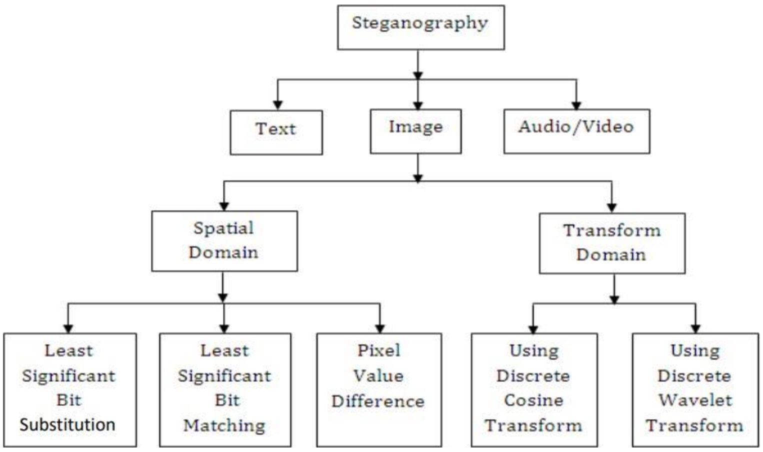

Steganography
Text
Image
Audio/Video
Spatial Domain
Transform Domain
Least Significant Bit Substitution
Least Significant Bit Matching
Pixel Value Difference
Using Discrete Cosine Transform
Using Discrete Wavelet Transform

Fig6.1: Block diagram of Steganography

As the above explanation goes, every steganography consists of three components:

1. Cover object
2. Message object
3. Resulting Steganographic object

In this project LSB substitution method is implemented and DCT method is discussed for image steganography. MATLAB is used for coding. The codes and result images are in the following report.

6.1.1. Technical Discussion:

In the current project image steganography is dealt with using data hiding.

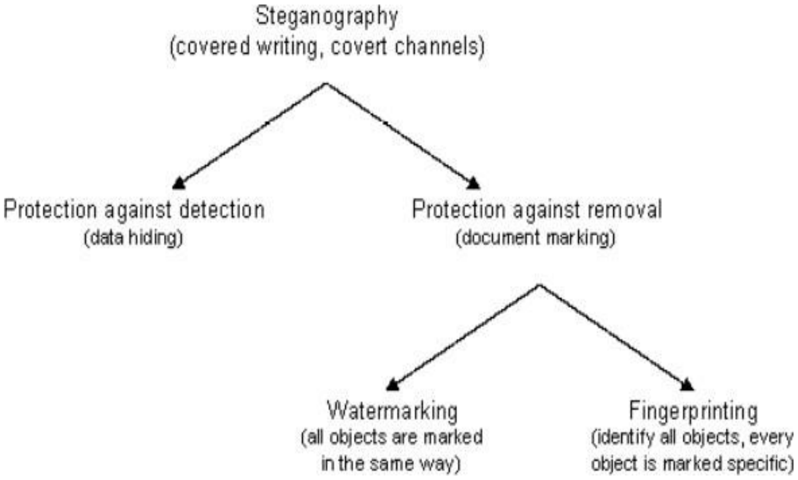

Steganography
(covered writing, covert channels)
Protection against detection
(data hiding)
Protection against removal
(document marking)
Watermarking
(all objects are marked in the same way)
Fingerprinting
(identify all objects, every object is marked specific)

Fig 6.2: Methods of Steganography

There are two different methods for image steganography:

1. Spatial methods

2. Transform methods

In spatial method, the most common method used is LSB substitution method.

6.1.2. Least significant bit (LSB):

LSB method is a common, simple approach to embedding information in a cover file. In steganography, LSB substitution method is used. I.e., since every image has three components (RGB). This pixel information is stored in encoded format in one byte. The first bits containing this information for every pixel can be modified to store the hidden text. For this, the preliminary condition is that the text to be stored has to be smaller or of equal size to the image used to hide the text. LSB based method is a spatial domain method.

But this is vulnerable to cropping and noise. In this method, the MSB (most significant bits) of the message image to be hidden are stored in the LSB (least significant bits) of the image used as the cover image. It is known that the pixels in an image are stored in the form of bits. In a grayscale image, the intensity of each pixel is stored in 8 bits (1byte).

Similarly, for a color (RGB-red, green, blue) image, each pixel requires 24 bits (8bits for each layer). The Human visual system (HVS) cannot detect changes in the color or intensity of a pixel when the LSB bit is modified. This is psycho-visual redundancy since this can be used as an advantage to store information in these bits and yet notice no major difference in the image.

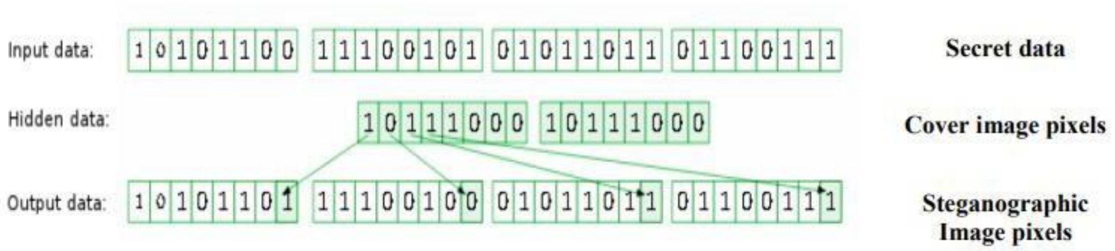


Fig 6.3: Least Significant Bit

6.1.3. Steps used in LSB steganography:

Steps for hiding message image:

1. Read the image to be used as cover image. Noise is added to make it easier to disguise changes due to embedding the message image.
2. Read the image to be sued as message image.
3. Separate the bit planes of each image. As it is known that the LSB (least significant bit) plane contains the least information associated with any image, and the MSB (most significant bit) plane contains most of the shape, color information of an image.

It is generally ideal to replace up to 4 least bit planes of the cover image, with the upper 4-bit planes without revealing changes in the resultant image. Lesser number of bit planes from the message image could be used, but the retrieved image would become distorted and loses information.

4. Replace the least 4-bit planes of cover image with the 4 most significant bit planes from message image.

5. Get the resultant Steganographic image by recombining these bit planes.

Retrieving message image:

1. Read the Steganographic image.
2. Extract the required number of bit planes of the image.
3. Recombining the lower four-bit planes would give the retrieved message image.

6.2. Discrete Cosine Transform (DCT) method:

When information is embedded in spatial domain, losses can occur such as when the image is cropped etc. To overcome this problem the information is embedded in frequency domain in such a way that we embed the secret information in the significant frequency values and omit the higher frequency part. First the required transformations are applied and then accordingly to hide the secret message, the transform coefficients are changed. Like in other transforms, decorrelation of the image data is required after applying discrete cosine transform (DCT). And encoding can be then done independently for each coefficient. Hence, compression efficiency is not lost.

In blocking method, blocks of the image are considered and DCT (discrete cosine transform) is done in order to break them. Each block is then subdivided into 64 parts (DCT coefficients). These coefficients are modified i.e., the color gets modified a little by storing some text or another image in it. Embedding the secret data in the carrier image is generally done for the DCT coefficients that are lower than the chosen threshold value. But embedding information in DCT coefficient value 0 is avoided as this may lead to visual distortion of the cover image.

6.3. Palette Modification:

In palette modification, the unused colors in an image’s color palette are replaced with colors to represent hidden message.

Palette Modification replaces the unused colors within an image’s color palette with colors that represent the hidden message.

For example, we have an image containing 6shades of blue and 5 shades of brown. By modifying the bits, it is possible to generate a completely new palette of colors that were originally absent in the previous image. This changed color palette may not be detected easily by human eye (HVS) and hence can be used to store other data or information.

- LSB technique can be used for BMP (bitmap) images. Since these involve lossless compression techniques.
- Blocking method – DCT and DWT – used for JPEG images. JPEG images have a lossy compression format, so spatial methods are unsuitable to perform steganography. DCT can be used to perform steganography on these images as, they undergo 2 layers of compression. One is lossless and then Huffman coding is used. The encryption data can be placed between these two layers.
- Palette based method – used for GIF images. GIF images have a very limited color palette. Therefore, palette modification method is more suitable.

CHAPTER-7

7.1. Password based Encryption (PBE):

PBE with MD5 and DES is a secure cipher class provided by JAVA itself. This project makes use of the service provided by JAVA. A small function calls for encryption and decryption return the encrypted and decrypted file by directing the input respectively. Here PBE is used with MD5 and DES. An 8-character password is given as input to this algorithm. The MD5 hashing generates a 128-bit stream with the given input (password) [5] [6] [13].

This 128-bit stream is used as key input which is necessary for the DES encryption as explained in literature review. Therefore, encryption is carried out in the form of blocks which is called block ciphering based upon a password.

7.2. Embedding:

Embedding is a process that inserts the bits of information into the byte array stream of cover image. The following describes the process.

Insertion of data should not change the information. That means, after inserting information into the cover image, and when the information is retrieved by receiver from image, the retrieved information should be same as the inserted information. The information to be inserted is encrypted twice before insertion, so the double encrypted information is processed first in order to keep the information consistent. The main problem could be only with the characters followed by symbol '\'. Some of the characters followed by back slash '\' could form an escape sequence character. To remove this, affect a single backslash is replace by multiple backslashes.

The cover image is processed and interpreted in the form of bytes. That is, information of each pixel of the cover image is stored in the form of a byte for corresponding RGB values. The last bit of each interpreted byte of all the pixels is replaced with one bit of information that is to be inserted. In the same way, each and every bit of information that are be hidden is inserted into available pixels of the cover image till the end of information.

Threshold:

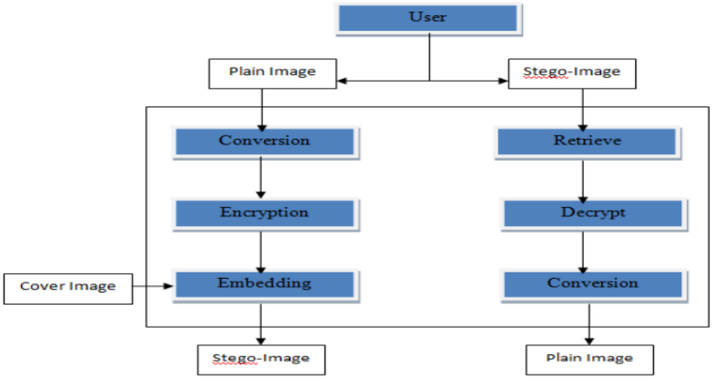
The insertion of bits of information is not straight-forward, but is done by selecting appropriate pixels to insert. That means, a threshold value is given as input by the user. And the pixel that satisfies the threshold (pixels value is greater than the threshold) is eligible for insertion. Before checking the threshold value of a pixel, its position is also considered to be even in terms of row and the column index. Finally, any pixel that is in an even position and whose value is greater than the user supplied threshold value is used for data to be inserted in it.

7.3. System Architecture:

7.3.1. Architecture:

Figure represents the architecture that is implemented. The modules of the steganographic tool are also included in the architecture. The user can either be the sender or the receiver.

Figure 7.1: System Architecture



A user is able to do all the operations as shown in figure 4 manually or automatically by using auto-mode. Both the manual and auto-mode follows the same system flow.

7.3.2. Flow of Execution

Figure shows the flow chart with the encryption part of the tool. The purpose of this project is to hide an image in other image, so a secret image would be an input. At first, the secret image is converted to a text file using Base 64 conversion. Then the generated text file is encrypted with a password-based encryption algorithm to generate an encrypted text file called ciphertext. Using a customized embedding algorithm, ciphertext is embedded on to a cover image. The output is the stego gramme (a cover image with a secret message embedded in it).

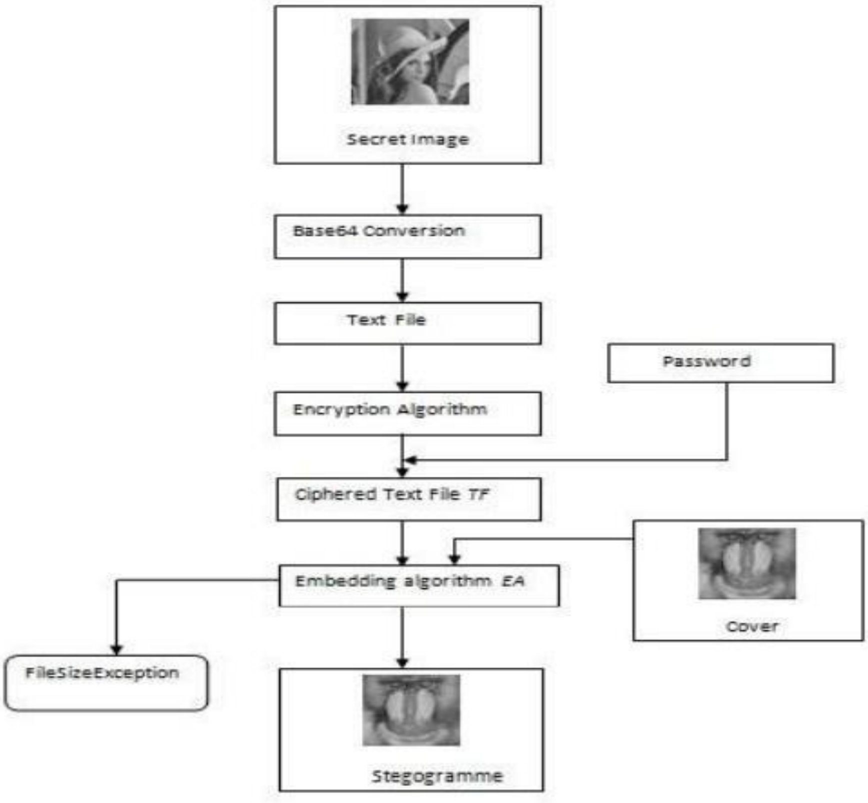


Figure 7.2: Flow of execution of sender’s operations.

Figure shows a flow chart with the decryption part of the project. To read the hidden message (secret image), the stego gramme has to be decrypted. So, the stego gramme is used as input to the retrieving algorithm. If the retrieving algorithm is not the same as the embedding algorithm, there is no way that the correct output can be obtained. The correct output from retrieving algorithm is the ciphertext is used as input to the decryption algorithm. This decryption algorithm is the same as the encryption algorithm; otherwise, the secret message cannot be determined. And also, the decryption algorithm takes a key (password) to generate plaintext.

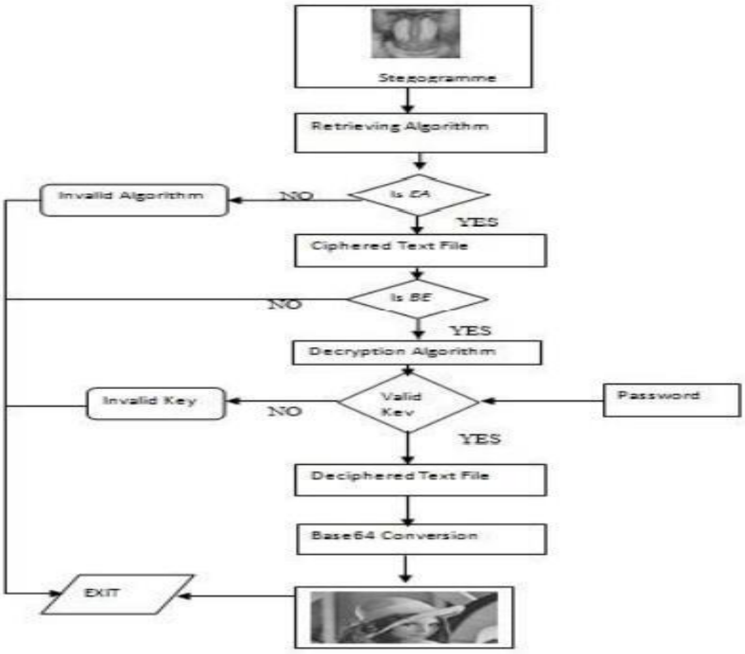


Figure 7.3: Flow of execution of receiver’s operations.

The correct password has to be used to get plaintext. Then, that plaintext is given as input to the Base64 converter to reconstruct the secret image for the plaintext.

7.3.3. Functional and Non-Functional Requirements:

The ‘functional requirements ‘satisfied by the tool are as follows:

1. Main window is the start of the application. Using this, a user can perform

   individual operations like encoding and decoding applicable at various levels in separate manner or individually.
2. Encoding tab leads the user to three windows that are used to perform 26 cryptographic and steganographic operations on images. A user can transform the secret image into a text file and encrypt the text file based upon a password and he can also embed the encrypted file into a cover image. The entire file generated by the user can be saved on desired location in a computer on which the tool is running.

iii. Decode tab is the tab where a receiver can generate his secret message. The embedded image can be processed here. With this tool a user can retrieve the information embedded in the image. The user can also decrypt an encrypted file with the help of a password. Finally, he will also be able to generate the secret image from a text file.

Among many 'non-functional requirements', the project must be very user-friendly. People will likely use such software tools with just a little more knowledge in using computers other than for everyday tasks. The developed steganographic tool satisfies all the above requirements.

Structural Requirements:

The tool satisfies the following structural requirements.

The project is to provide user-friendly operational steganographic tool for the multilevel protection. This tool allows users to operate easily on images and files with different formats like.txt, png, enc etc.

1. It gives Self-explanative user interface.

2. On clicking encode tab, all the encoding options appear.

3. On clicking decode tab, all the decoding options should appear.

4. All the buttons that are used to instruct the tool should appear.

When a user closes the main window, all operational windows must stop their action and close.

Encode tab:

1. A user can generate a text format of an image file.

2. After generating a text file, a user can encrypt the text file with a password.

3. Finally, a user can embed the encrypted file into a cover image, with format of the image being PNG so that it will be easy to transfer in any network.

4. A user is allowed to save all the output files in the hard disk.

Decode tab:

1. A user is allowed to load stego-image to retrieve hidden information.
2. A user is allowed to decrypt the encrypted file using same password that was used at the time of encryption.
3. At the end, a user can generate an image from the retrieved information.

Auto mode tab:

1. The proposed operations, either encoding or decoding can be performed automatically without going through all the steps.
2. All the encoding and decoding operations are integrated into two individual windows i.e., encode and decode respectively.
3. All the inputs are given at a time both for encoding and decoding operations.

CHAPTER-8

8.1. Analysis:

The important task of this project is to embed an image into an image, not directly but by providing security through multiple levels. With the proposed system code, the requirement is fulfilled with any digital format of input. All the inputs can be transformed to appropriate format and any information can be embedded into a cover image. After embedding, they are automatically stored in portable network format for sharing purpose on network. Hence the input image can be of any format (jpg, jpeg, png, gif) for encoding and gives out a cover image with input embedded in it. The receiver's side of this tool takes the cover image as input. By applying appropriate operations on the cover image, the secret digital image is given as an output.

8.2. Design:

Users can interact with the system through interface. A user can either be a sender or a receiver. therefore, all the conversion, cryptographic and steganographic operations can be performed by the user.

Embed Sequence Diagram:

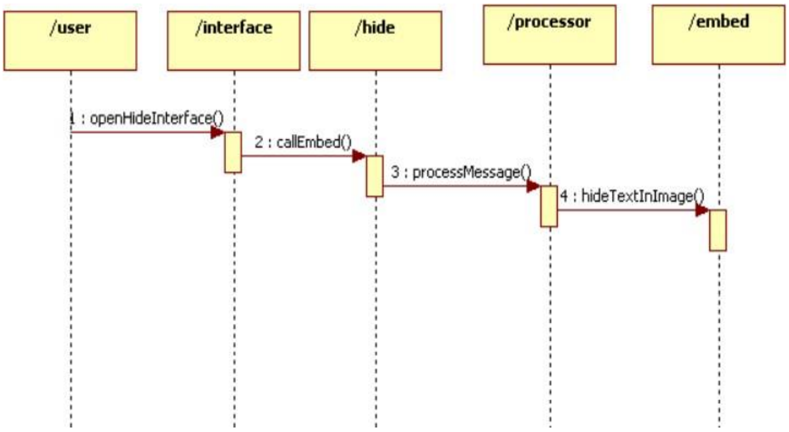


Figure 8.1: Sequence diagram for embedding process.

Figure is a sequence diagram showing the flow of instructions in the process of embedding. A user interacts with the system to embed the message in cover image. The embed function invoked by the user’s operation processes the input message in the form of bits information. This information in the form of a bit stream is inserted into the bytes of the cover image by processing the cover image.

CHAPTER-9

9.1. RESULTS:

- Image Steganography refers to the process of hiding data within an image file. The image selected for this purpose is called the cover-image

Fig. 9.1.: The original cover image

- A secret image is first processed into n-shares which are then hidden in n-user selected different cover images. Now select the source file in which you want to hide the secret message, and then choose the file to hide or write the text message to disappear.

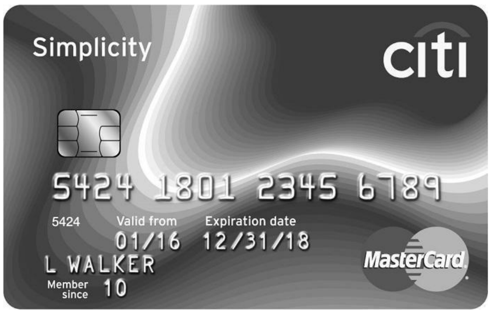


Fig. 9.2: The secret message image to be hidden: (converted to greyscale)

- A secret message is to be embedded in another cover object which may not be of any significance in such a way that the encrypted data would finally display only the cover data. So, it cannot be detected easily to be containing hidden information unless proper decryption is used.

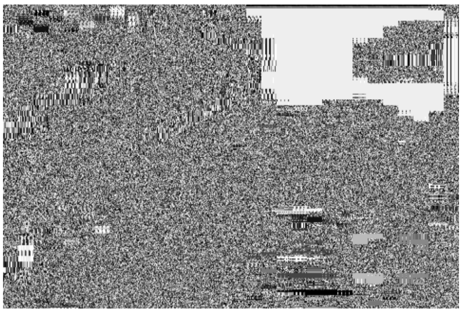
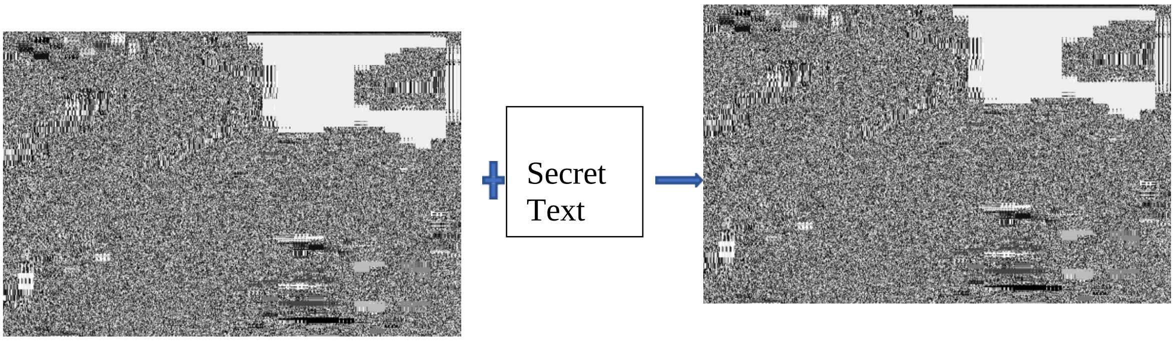


Fig 9.3: Encrypted Secret Image

- For hiding secret message in a pixel, the physical location of a pixel is considered and then the binary format of that pixel's value is used to hide the secret message. The most common method for steganography in image is LSB insertion method.



Encrypted Secret Image

Encrypted Image + Secret Text in it

Fig. 9.4: Hiding of secret key inside the encrypted secret image

- The encrypted text from Secret image and a cover text are taken as inputs and white space text steganography technique is used to hide the encrypted text in the whitespaces present in cover text.

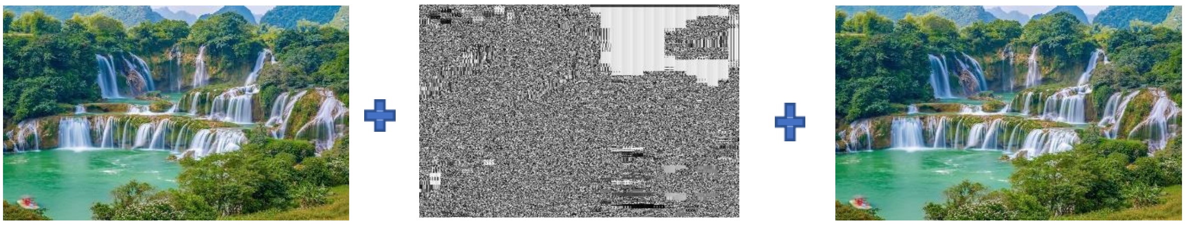


Fig 9.5: Hiding of encrypted image containing the secret text inside the cover image

- The message segmentation LSB image steganography technique was suggested here by splitting the long secret message into number of short segments. Then hide these short segments in different parts of the best matched LSB in the pixels of the stego-image.

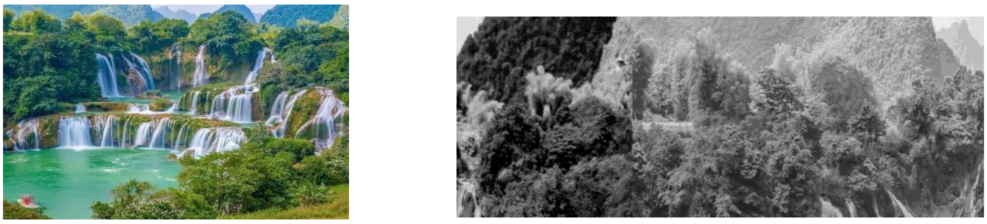
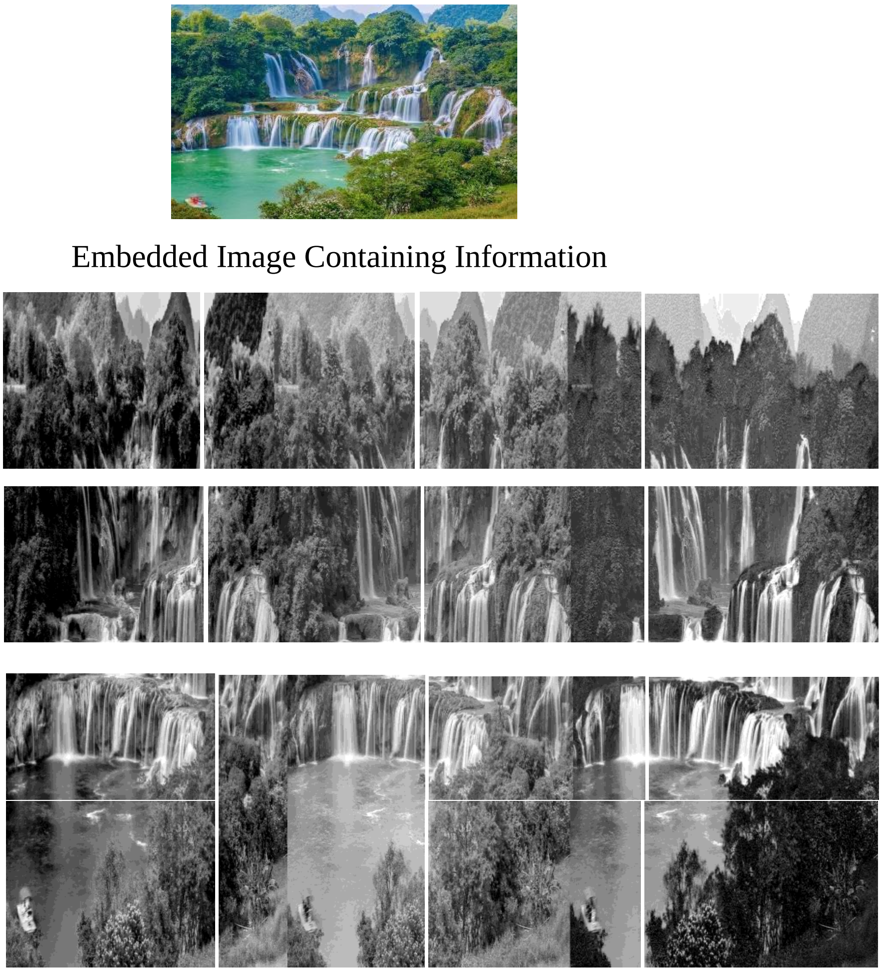


Fig 9.1.6: Segmentation of the stego image

- Image stitching is concerned with combining two or more images of the same scene into one high resolution image. Then the cover image is decomposed into some non-overlapped sub-images. After each sub-image is embedded.

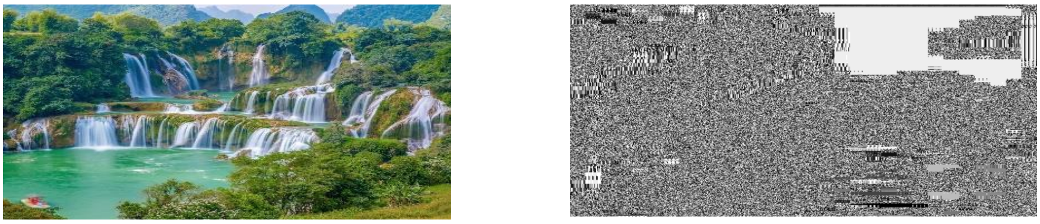


Embedded Image Containing Information

Sub Images

Fig 9.7: Stitching of sub images into a single image

- Least Significant Bits (LSB) of the cover image using stego key. At the recipient's end, first the secret message is extracted using stego key then decrypted using cipher key. The cipher key for message encryption is 256 bits long. Encoder replaces the N Least Significant Bits of the pixel value with the same number of most significant bits from the secret pixel value.

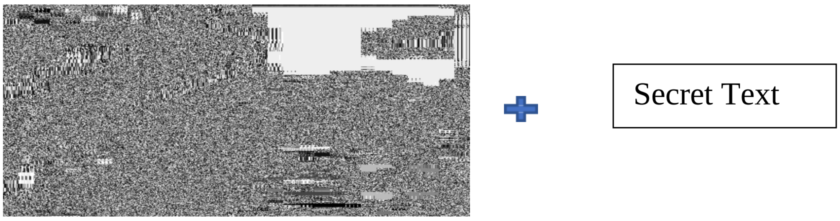


Merged Image Containing Information

Encrypted Image + Secret Text in it

Fig 9.8: Extracting the encrypted image from the least significant bits of the stego image.

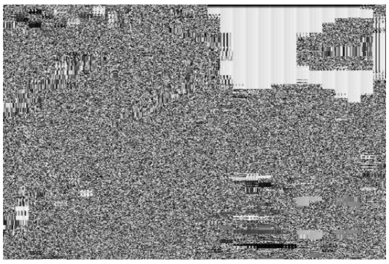
- A number of horizontal and vertical blocks at the sender side will be generated, and then mixed with the encrypted image before transmitting it to the receiver. The receiver will need this information to reconstruct the same secret transformation table after extracting the secret information from the encrypted image.



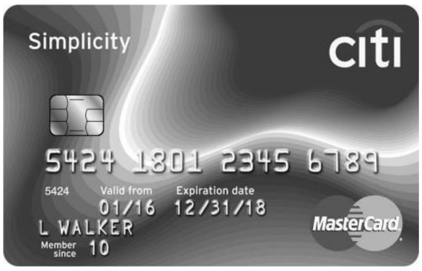
Encrypted Image + Secret Text in it

Fig 9.9: Extracting the secret text from the encrypted image

- The revers process that of the encryption process. Which consist of text as the input, the key is same for decryption process which for encryption.



Encrypted Image



Decrypted Image

Fig 9.10: Decrypting the encrypted image using AES Algorithm

9.2. Conclusion

The developed steganographic tool is used to encrypt and decrypt the image. In this project, security to confidential data is achieved through multiple levels with the combination of both cryptographic and steganographic strategies. In the process of embedding information into the cover image, a successful threshold strategy is used. A bit of information is inserted into a pixel only if the pixel satisfies threshold value and position constraint. The embedding image can be of any format (jpeg, pig, gif, png). The generated stego-image is in .png format because the image quality of this format is reasonable with the file size.

It is observed that through LSB Substitution Steganographic method, the results obtained in data hiding are pretty impressive as it utilizes the simple fact that any image could be broken up to individual bit-planes each consisting of different levels of information. It is to be noted that as discussed earlier, this method is only effective for bitmap images as these involve lossless compression techniques. Also, in this project grey-scale images have been used for demonstration. But this process can also be extended to be used for color images where, bit plane slicing is to be done individually for the top four bit-planes for each of R, G, B of the message image, which are again to be placed in the R, G, B planes of the cover image, and extraction is done similarly.

It can be observed that the result image after extraction is not very clear as the initial message/secret image. This can be explained by the fact that this is a very high-resolution image. From Fig. 8, it is seen that data is visible in the message image in planes 5, 6, 7 to the human eye, and rest of the bit-planes appear to be dark/ black, but these also have tiny bits of information which is ignored for the process of data hiding. Better results can be obtained if the message image to be hidden is of a lower resolution. This is observed from Image set 2, where the extracted image has less loss compared to the original message image.

It is also important to discuss that though steganography was once undetected, with the various methods currently used, it is not only easy to detect the presence but also retrieving them is easier. For instance, without having to use a software or complex tools for detection, simple methods to observe if an image file has been manipulated are:

1. **Size of the image:** A Steganographic image has a huge storage size when compared to a regular image of the same dimensions. i.e., if the original image storage size would be few KBs, the Steganographic image could be several MBs in size. This again varies with the resolution and type of image used.
2. **Noise in image:** A Steganographic image has noise when compared to a regular image. This is the reason why initially little noise is added to the cover image, so that the Steganographic image doesn't appear very noisy when compared to the original cover image.

Though this project focusses on LSB and spatial domain steganography, few details about transform domain methods have also been researched, basics of which have been discussed. So, through the various articles and theory available, it is observed that transform domain methods perform better in comparison with spatial domain methods.

9.3. Future Scope:

As a part of security, the pixels of the cover image are filtered both according to their position and the threshold limit. Because of this, the space availability of data insertion could become very less. Therefore, the embedding information should be small for successful embedding. New ideas could be developed on increasing the space availability in the cover image to insert as much data as possible.

1. Steganography can protect data by hiding it but using it alone may not guarantee total protection.
2. In case of encryption, by seeing the meaningless appearing sequence of bits enemy can detect that some illegal message is being sent.
3. However, if one uses both methods, this will lead to ‘security in depth’.
4. The message should first be encoded using a strong encryption algorithm and then embedded into a carrier.

References:

[1]. Rahul Kumar, Ajith Pratap Singh, Arun Kumar Shukla, Rishabh Shukla “Enhancing Security using Image Processing” Sam Higginbottom Institute of Agriculture Technology and Sciences, Allahabad, India | International Journal of Innovative Research in Science, Engineering and Technology - Vol. 4, Issue 4, April2015.

[2]. Jyoti Ka Kapoor, Akshay. J. Barger “Security using image processing” K.J. Somaiya College of Engineering, Mumbai, India | International Journal of Managing Information Technology (IJMIT) - Vol. 5, No. 2, May2013

[3]. WIKI “Steganography”

[4]. WIKI “Advanced Encryption Standard”

[5]. WIKI “Cryptography”

[6]. Roshni Pedate, Amana Patel “Image Encryption and Decryption using AES Algorithm” Fr. Concepcion Rodrigues College of Engineering, Mumbai, India | IJECT – Vol. 6, Issue 1, January (2015), pp.23-29

[7]. Champ kamala B S, Padmini K, Radhika D K “Least Significant Bit algorithm for image steganography” Don Bosco Institute of Technology, Bangalore, India | International Journal of Advanced Computer Technology (IJACT) Nikon | Imaging Products | Still Images – Nikon D7100 “http://imaging.nikon.com/lineup/dslr/d7100/ sample.htm”

[8]. Credit Card – Citi Simplicity – citi.com “https:// www.citi.com/credit-cards/credit- card details/Citi. Action ID=Citi-simplicity-credit-card”.

Appendix

List of definition used in the project:

1. **Carrier**: The file or signal use to hide and transmit the payload. (see cover)
2. **Channel:** The input into the steganography system, such as an image.
3. **Cover:** The media object in which the hidden message (payload) will be transmitted. "Cover" refers to this piece of media before it is embedded with the secret data. Such images are specifically called "cover images" and texts are specifically called "cover texts"
4. **Double-stagging:** A specific steganography-jamming technique that involves applying steganography to a media object thought to be a stego-object with the intention of destroying the originally embedded message.
5. **Plaintext**: An unembedded, unencoded message.
6. **Payload:** The message to be transmitted.
7. **Stego-object:** The media object after the hidden message has been embedded within it. Called "stego text" when embedded in text and "stego image" when embedded in an image.
8. **Steganography-jamming:** A general stego analysis tactic whereby the stego analysist seeks to destroy the hidden message without actually uncovering it.