A Mini Project report on

SIMPLE AND SECURE IMAGE STEGANOGRAPHY USING LSB AND TRIPLE XOR OPERATION ON MSB

Submitted

In partial fulfillment of the Requirement for the award of the Degree of

BACHELOR OF TECHNOLOGY

In

ELECTRONICS AND COMMUNICATION ENGINEERING

By

S. DILEEP REDDY 19W91A04L0

V. NAGARAJU 19W91A04P5

T. NIRAJ GOUD 19W91A04M4

Under the Esteemed Guidance of

Mrs. Jhansi Rani

Associate Professor

Department of Electronics & Communication Engineering

MALLA REDDY INSTITUTE OF ENGINEERING AND TECHNOLOGY

Approved by A.I.C.T.E. - New Delhi, Affiliated to J.N.T. University Hyderabad

Maisammaguda, Dhulapally, Secunderabad-500 100

**December-2022**

 Malla Reddy Institute of Engineering and Technology

(Sponsored by Malla Reddy Educational Society)

 ISO 9001-2008 Certified institution, Affiliated to JNTU, Hyderabad Maisammaguda, Dhulapally (Post via Hakimpet), Sec’Bad - 500 100.

Department of Electronics and Communication Engineering

CERTIFICATE

This is to certify that the mini project entitled “SIMPLE AND SECURE IMAGE STEGANOGRAPHY USING LSB AND TRIPLE XOR OPERATION ON MSB” that is being submitted by S. DILEEP REDDY (19W91A04L0), V. NAGARAJU (19W91A04P5), T. NIRAJ GOUD (19W91A04M4) under the guidance of Mrs. Jhansi Rani for the award of B.Tech Degree in ELECTRONICS AND COMMUNICATION ENGINEERING from the MALLAREDDY INSTITUTE OF ENGINEERING & TECHNOLOGY, Maisammaguda (Affiliated to JNTU Hyderabad) is a record of Bonafide work carried out by them under our guidance and supervision. The results embodied in this mini project have not been submitted to any other university or institute for the award of any degree.

Project Guide HOD

Mrs. JHANSI RANI Dr. P.SAMPATH KUMAR

Department of ECE Department of ECE

 Malla Reddy Institute of Engineering and Technology

(Sponsored by Malla Reddy Educational Society)

 ISO 9001-2008 Certified institution, Affiliated to JNTU, Hyderabad Maisammaguda, Dhulapally (Post via Hakimpet), Sec’Bad - 500 100.

Department of Electronics and Communication Engineering

DECLARATION

We S. DILEEP REDDY (19W91A04L0), V. NAGARAJU (19W91A04P5), T. NIRAJ GOUD (19W91A04M4) hereby declare that the mini project entitled “SIMPLE AND SECURE IMAGE STEGANOGRAPHY USING LSB AND TRIPLE XOR OPERATION ON MSB” is bonafide work done and submitted under the guidance of Mrs. M. SRAVANTHI in partial fulfillment of the requirements for the award of the degree of BACHELOR OF TECHNOLOGY in ELECTRONICS AND COMMUNICATION ENGINEERING.

DEPARTMENT OF ECE

S. DILEEP REDDY 19W91A04L0

V. NAGARAJU 19W91A04P5

T. NIRAJ GOUD 19W91A04M4

ACKNOWLEDGEMENT

We are very much thankful to Director, **Shri. P.PRAVEEN REDDY** for giving us this opportunity to do this mini project. We express our deep sense of gratitude to him for his constant guidance and inspiring words.

We express our profound thanks to our Principal, **Dr. M. ASHOK,** for extending all the college facilities for the completion of the mini project.

We would like to thank **Dr. P. SAMPATH KUMAR** Professor and Head of the Department of Electronics and Communication Engineering & **Dr. RAJSREE RAO** Dean of Academics for having provided the freedom to use all the facilities available in the department, especially the laboratories and the library, at anytime.

We feel highly obliged to our project coordinator **Mr. SHAIK SOHEL PASHA (Ph.D)**, Asst Professor and project guide **Mrs.JHANSI RANI , (Ph.D)**, Associate Professor Department of Electronics and Communication Engineering for their constant encouragement and moral support. They have been a source of valuable guidance, suggestions and kindness during the course of the project work. We find no words to express our gratitude and thanks to them.

We sincerely thank all the staff of the Department of Electronics and Communication Engineering, for their timely suggestions, healthy criticism and motivation during the course of our study. We would also like to thank our friends for always being there to provide required help or support. With great respect and affection, we thank our parents who were the backbone behind our deeds.

Finally, we express our immense gratitude with pleasure to one and all who have either directly or indirectly contributed to our need at right time for the development and execution of project work.

S. DILEEP REDDY 19W91A04L0

V. NAGARAJU 19W91A04P5

T. NIRAJ GOUD 19W91A04M4

**CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **TOPIC** | **PAGE NO.** |
|  | CONTENTS | i |
|  | LIST OF FIGURES | iv |
|  | LIST OF TABLES |  |
|  | ABSTRACT | 1 |
| **CHAPTER - 1** | **INTRODUCTION** | 2 |
|  | **1.1** Steganography | 2 |
|  | **1.2** Implementation | 4 |
|  | **1.3** Evolution of steganography code breakers | 5 |
|  | **1.4** Types of steganography message in text | 6 |
|  | **1.5** Dissecting steganography | 7 |
|  | **1.6** Discrete cosine transform (dct) method | 9 |
|  | **1.7** Palette modification | 10 |
|  | **1.8** Steganography security | 10 |
|  | **1.9** Techniques for image steganography | 12 |
|  | **1.10** Image steganography | 16 |
|  | **1.11** Characterizing data hiding techniques | 16 |
|  | **1.12** Binary representation of an RGB color image | 17 |
|  | **1.13** Algorithms used in proposed method | 17 |
|  | **1.13.1** RSA Algorithm | 18 |
|  | **1.13.2** Implementation of LSB | 19 |

|  |  |  |
| --- | --- | --- |
|  | **1.13.3** Application of steganography | 23 |
|  | **1.14** LSB encoding | 26 |
| **CHAPTER - 2** | **LITERATURE SURVEY** |  |
| **CHAPTER - 3** | **INTRODUCTION TO MATLAB** | 27 |
|  |  |  |
|  | **2.1** Introduction to matlab | 27 |
|  | **2.2** The Matlab system | 28 |
|  | **2.3** Graphical user interface (GUI) | 29 |
|  | **2.4** Getting started | 30 |
|  | **2.5** Development environment | 31 |
|  | **2.5.1** Introduction | 31 |
|  | **2.5.2** Starting matlab | 31 |
|  | **2.5.3** Quitting matlab | 31 |
|  | **2.5.4** Matlab desktop | 31 |
|  | **2.5.5** Desktop tools | 32 |
| **CHAPTER - 4** | **IMAGE STEGANOGRAPHY TECHNIQUES** | 36 |
|  |  |  |
|  | **3.1** Spatial domain technique | 36 |
|  | **3.2** Masking and Filtering | 36 |
|  | **3.3** Transform and domain technique | 37 |
|  | **3.4** Distortion technique | 37 |
|  | **3.5** Characteristics feature of data hiding techniques | 38 |
|  | **3.6** Image steganalysis | 38 |
|  | **3.6.1** Image steganalysis | 39 |
|  | **3.6.2** Steganalysis tools | 39 |
|  | **3.7** Least significant bit (LSB) techniques | 39 |
|  | **3.8** Cover and hidden images | 40 |

|  |  |  |
| --- | --- | --- |
|  | **3.9** Data embedding | 41 |
|  | **3.10** Data extraction | 42 |
|  | **3.11** Image encoding algorithm | 43 |
|  | **3.12** Pseudo-random encoding technique | 44 |
|  | **3.12.1** Embedding algorithm | 44 |
|  | **3.12.2** Extraction of hidden image | 45 |
|  | **3.12.3** Message extraction algorithm | 45 |
|  | **3.13** Gray scale | 46 |
|  | **3.14** Least significant bit technique | 49 |
| **CHAPTER - 5** | Design and implementation | 54 |
| **CHAPTER - 6** | Experiments and results | 57 |
| **CHAPTER - 7** | Conclusion & Future Scope | 60 |
|  | References | 60 |
|  | Code |  |

**LIST OF FIGURES**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **TITLE** | | **PAGE NO.** |
| **1.** | FIG 2.1 | Matlab procedure | 29 |
| **2.** | FIG 3.1 | LSB insertion technique | 41 |
| **3.** | FIG 3.2 | LSB extraction technique | 41 |
| **4.** | FIG 3.3 | Gray scale color | 46 |
| **5.** | FIG 3.4 | Steganography block diagram | 47 |
| **6.** | FIG 3.5 | Algorithm of least significant bit | 51 |
| **7.** | FIG 5.1 | Embedded scheme | 54 |
| **8.** | FIG 5.2 | Extraction scheme | 55 |
| **9.** | FIG 6.1 | Original image | 57 |
| **10** | FIG 6.2 | Message image used | 58 |
| **11.** | FIG 6.3 | Output of encryption | 58 |
| **12.** | FIG 6.4 | Output of decryption | 59 |

**LIST OF ABBREVATIONS**

|  |  |
| --- | --- |
| **LSB** | Least Significant Bit. |
| **MSB** | Most Significant Bit. |
| **HVS** | Human Visual System |
| **GUI** | Graphical User Interfaces |
| **DFT** | Discrete Fourier Transformation Technique |
| **DCT** | Discrete Cosine Transformation Technique |
| **DWT** | Discrete Wavelet Transformation Technique |

**SECURE IMAGE STEGANOGRAPHY USING LSB AND TRIPLE XOR OPERATION ON MSB**

**ABSTRACT**

Least Significant Bit (LSB) is a very popular method in the spatial domain of steganographic images. This method is widely used and continues to be developed to date, because of its advantages in steganographic image quality. However, the traditional LSB method is very simple and predictable. It needs a way to improve the security of hidden messages in this way. This research proposes a simple and safe way to hide messages in LSB techniques. Three times the XOR operation is done to encrypt the message before it is embedded on the LSB. To facilitate the process of encryption and decryption of messages, three MSB bits are used as keys in XOR operations. The results of this study prove that this method provides security to messages with very simple operation. The imperceptibility quality of the stego image is also excellent with a PSNR value above 50 dB.

**CHAPTER-1**

* 1. **STEGANOGRAPHY**

**INTRODUCTION**

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, and 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 Steganalysis.

Steganography is of different types:

1. Text steganography
2. Image steganography
3. Audio steganography
4. 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 refers to the science of “invisible” communication. Unlike

Cryptography, where the goal is to secure communications from an eaves-dropper, steganographic techniques strive to hide the very presence of the message itself from an observer. The general idea of hiding some information in digital content has a wider class of applications that go beyond steganography, The techniques involved in such applications are

collectively referred to as information hiding. For example, an image printed on a document could be annotated by metadata that could lead a user to its high resolution version. In general, metadata provides additional information about an image. Although metadata can also be stored in the ﬁle header of a digital image, this approach has many limitations. Usually, when a ﬁle is transformed to another format (e.g., from TIFF to JPEG or to BMP), the metadata is lost. Similarly, cropping or any other form of image manipulation destroys the metadata. Finally, metadata can only be attached to an image as long as the image exists in the digital form and is lost once the image is printed. Information hiding allows the metadata to travel with the image regardless of the ﬁle format and image state (digital or analog).

A special case of information hiding is digital watermarking. Digital watermarking is the process of embedding information into digital multimedia content such that the information (the watermark) can later be extracted or detected for a variety of purposes including copy prevention and control. Digital watermarking has become an active and important area of research, and development and commercialization of watermarking techniques is being deemed essential to help address some of the challenges faced by the rapid proliferation of digital content. The key diﬀerence between information hiding and watermarking is the absence of an active adversary. In watermarking applications like copyright protection and authentication, there is an active adversary that would attempt to remove, invalidate or forge watermarks.

In information hiding there is no such active adversary as there is no value associated with the act of removing the information hidden in the content. Nevertheless, information hiding techniques need to be robust against accidental distortions. Unlike information hiding and digital watermarking, the main goal of steganography is to communicate securely in a completely undetectable manner. Although steganography is an ancient art, ﬁrst used against the persian by the Romans, it has evolved much through the years.

Image Steganography has many applications, especially in today’s modern, high tech world. Privacy and anonymity is a concern for most people on the internet. Image Steganography allows for two parties to communicate secretly and covertly. It allows for some morally-conscious people to safely whistle blow on internal actions; it allows for

copyright protection on digital files using the message as a digital watermark. One of the other main uses for Image Steganography is for the transportation of high-level or top-secret documents between international governments. While Image Steganography has many legitimate uses, it can also be quite nefarious. It can be used by hackers to send viruses and trojans to compromise machines, and also by terrorists and other organizations that rely on covert operations to communicate secretly and safely .

* 1. **IMPLEMENTATION**

There are currently three effective methods in applying Image Steganography: LSB Substitution, Blocking, and Palette Modification. LSB (Least Significant Bit) Substitution is the process of modifying the least significant bit of the pixels of the carrier image. Blocking works by breaking up an image into “blocks” and using Discrete Cosine Transforms (DCT). Each block is broken into 64 DCT coefficients that approximate luminance and color—the values of which are modified for hiding messages. Palette Modification replaces the unused colors within an image’s color palette with colors that represent the hidden message. With LSB Substitution I could easily change from Image Steganography to Audio Steganography and hide a zip archive instead of a text message.

LSB Substitution lends itself to become a very powerful Steganographic method with few limitations. LSB Substitution works by iterating through the pixels of an image and extracting the ARGB values. It then separates the color channels and gets the least significant bit. Meanwhile, it also iterates through the characters of the message setting the bit to its corresponding binary value 3 .

Steganography can be viewed as akin to cryptography. Both have been used throughout recorded history as means to protect information. At times these two technologies seem to converge while the objectives of the two differ. Cryptographic techniques "scramble" messages so if intercepted, the messages cannot be understood. Steganography, an essence, "camouflages" a message to hide its existence and make it seem "invisible" thus concealing the fact that a message is being sent altogether. An encrypted message may draw suspicion

while an invisible message will not. In an ideal world we would all be able to openly send encrypted email or files to each other with no fear of reprisals.

However there are often cases when this is not possible, either because you are working for a company that does not allow encrypted email or perhaps the local government does not approve of encrypted communication (a reality in some parts of the world). This is where steganography can come into play. A good steganography system should fulfill the same requirements posed by the "Kerckhoff principle" in cryptography. This means that the security of the system has to be based on the assumption that the "enemy" has full knowledge of the design and implementation details of the steganographic system.

The only missing information for the "enemy" is a short easily exchangeable random number sequence, the secret key, and without the secret key, the "enemy" should not have the slightest chance of even becoming suspicious that on an observed communication channel hidden communication might take place. Steganography cannot be detected. Therefore, it is used when encryption is not permitted. Or, more commonly, steganography is used to supplement encryption. An encrypted file may still hide information using steganography, so even if the encrypted file is deciphered, the hidden message is not seen.

# EVOLUTION OF STEGANOGRAPHY CODE BREAKERS

David Kahn's The Code breakers and Bruce Norman’s Secret Warfare: The Battle of Codes and Ciphers recounts numerous tales of steganography .

**INVISIBLE LINK :** An innocent letter may contain a very different message written between the lines with invisible ink. Common sources for invisible inks are milk, vinegar, fruit juices and urine. All of these darken when heated. Later on, more sophisticated inks were developed which react to various chemicals.

**MICRODOTS:** The Germans developed microdot technology. Microdots are photographs the size of a printed period having the clarity of standard-sized typewritten pages. The first microdots were discovered masquerading as a period on a typed envelope carried by a German agent in 1941. The message was not hidden, nor encrypted. It was just

so small as to not draw attention to itself (for a while). Besides being so small, microdots permitted the transmission of large amounts of data including drawings and photographs.

# TYPES OF STEGANOGRAPHY MESSAGES IN TEXT

Secret messages can be hidden in text format by reframing the text of the carrier file, while maintaining the context. One form of steganography is a program called Spam Mimic. Based on a set of rules called a mimic engine by Peter Wayner, it encodes your message into what looks like your typical, quickly deleted Spam message. However, hiding a message in plain text is a thing of the past, as people are suspicious of irrelevant text. Messages in still images most popular tool is outguess. Messages in audio data are hidden in layer III of the encoding process of the MP3 file. Messages in audio are always sent along with ambient noise. The data is hidden in the heart of the layer III encoding process of the MP3 file, namely the inner loop during compression.

The inner loop limits the input data and increases the step size until the data can be coded with the available number of bits. The data is compressed, encrypted and then hidden in an MP3 bit stream. Messages in Video embedding information into multimedia data has gained increasing attention lately. The method of encryption is the same as in audio steganography. Video files are generally very good carrier files since they have a lot of irrelevant bits. An Example Fishing freshwater bends and saltwater coasts rewards anyone feeling stressed. Resourceful anglers usually find masterful leapers fun and admit swordfish rank overwhelming anyday. “Send lawyers guns and money” Steganography is closely related to the problem of "hidden channels" in secure operating system design, a term which refers to all communication paths that can not easily be restricted by access control mechanisms (e.g. two processes that communicate by modulating and measuring the CPU load).

Steganography is also closely related to spread spectrum radio transmission, a technique that allows to receive radio signals that are over 100 times weaker than the atmospheric background noise, as well as TEMPEST, techniques which analyze RF transmissions of computer and communication equipment in order to get access to secret

information handled by these systems. Most communication channels like telephone lines and radio broadcasts transmit signals which are always accompanied by some kind of noise. This noise can be replaced by a secret signal that has been transformed into a form that is indistinguishable from noise without knowledge of a secret key and this way, the secret signal can be transmitted.

# DISSECTING STEGANOGRAPHY

Steganography is a term used for hiding messages within an image. Any color pixel is made of a combination of red –green-blue mode (RGB) wherein each RGB component consists of 8 bits. If letters in ASCII are to be represented within the color pixels, the rightmost digit, called the least significant bit (LSB), can be altered. Any variation in the value of this bit leads to very minimal variation in color. If we have to hide the word ‘digit’ in the image, we take the LSB of every color and hide each bit of the word in its RGB combination. To insert the letter ‘D’ we modify three color pixels with three bits in each color pixel, we utilize 14 color pixels to hide the entire word with only 1 bit in the 14th pixel.

## Steps for hiding an an image using steganography :-

1. Start s-tool and window explorer using the later as drag and drop interface for the software.
2. Drag and drop the image to be used as the carrier file from the explorer onto the actions window in s-tool.
3. Drag and drop the data file on the carrier file.
4. Give a passphrase and encryption algorithm when prompted. Pass these to the receiver too.
5. The hidden file is ready. Receiver has to click on the “reveal” button to extract the data.

Steganography simply takes one piece of information and hides it within another. Computer files (images, sounds recordings, even disks) contain unused or insignificant areas of data. Steganography takes advantage of these areas, replacing them with information (encrypted mail, for instance). The files can then be exchanged without anyone knowing what really lies inside of them. An image of the space shuttle landing might contain a private letter to a friend. A recording of a short sentence might contain your company's plans for a secret

new product. Steganography can also be used to place a hidden "trademark" in images, music, and software, a technique referred to as watermarking.

There are two different methods for image steganography:

1. Spatial methods
2. Transform methods

In spatial methods, the most common method used is LSB substitution method. Least significant bit (LSB) method is a common, simple approach to embedding information

in a cover file. In steganography, the 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 colour (RGB-red, green, blue) image, each pixel requires 24 bits (8 bits for each layer). The Human visual system (HVS) cannot detect changes in the colour 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.

## Steps used in LSB steganography:-

1. **Steps for hiding message image :-**
   1. Read the image to be used as a cover image. Noise is added to make it easier to disguise changes due to embedding the message image.
   2. Read the image to be used as a message image.
   3. Separate the bit planes of each image.
   4. Replace the least 4 bitplanes of cover image with the 4 most significant bit planes from message image.
   5. Get the resultant Steganographic image by recombining these bitplanes.

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

## Retrieving message image:-

* 1. Read the Steganographic image.
  2. Extract the required number of bitplanes of the image.
  3. Recombining the lower four bitplanes would give the retrieved message image.

# DISCRETE COSINE TRANSFORM (DCT) METHOD

When information is embedded in the spatial domain, losses can occur such as when the image is cropped etc. To overcome this problem the information is embedded in the 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, de-correlation 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 the blocking method, blocks of the image are considered and DCT (discrete cosine transform) is done in order to break them. 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.

# PALETTE MODIFICATION

In palette modification, the unused colours in an image’s colour palette are replaced with colours to represent hidden messages. Palette Modification replaces the unused colours within an image’s colour palette with colours that represent the hidden message. For example, we have an image containing 6 shades of blue and 5 shades of brown. By modifying the bits, it is possible to generate a completely new palette of colours that were originally absent in the previous image. This changed colour palette may not be detected easily by the human eye (HVS) and hence can be used to store other data or information.

# STEGANOGRAPHIC SECURITY

In steganography, unlike other forms of communications, one’s awareness of the underlying communication between the sender and receiver defeats the whole purpose. Therefore, the first requirement of a steganographic system is its undetectability. In other words, a steganographic system is considered to be insecure, if the warden Wendy is able to differentiate between coverobjects and stego-objects. There have been various approaches in defining and evaluating the security of a steganographic system. Zollner et al. were among the first to address the undetectability aspect of stegano graphical systems.

They provide an analysis to show that theoretically secure steganography is possible if the embedding operation has a random nature and the embedded message is independent from both the cover-object and stego-object. That is, Wendy has no access to the statistics, distribution, or conditional distribution of the cover-object. Based on cryptographic principles, they propose the design of encryption-decryption functions for steganographic embedding and detection. In this setting, the underlying distribution of the cover-objects is known by the attacker, and undetectability is defined in a conditional sense as the inability of a polynomial-time attacker (Wendy) to distinguish the stego-object from a cover-object. This model assumes that the stego-object is a distorted version of the cover-object, however, it does not attempt to probabilistically characterize the stego object.

In the best case scenario, Wendy also knows the distribution of stego-objects and makes a decision by performing a binary hypothesis test. Consequently, the detectability of a stego system is based on relative entropy between the probability distributions of the cover-object and stego-object, denoted by Pc and Ps, respectively, i.e., D(Pc||Ps) = Z Pc log Pc Ps . From this equation, we note that D(Pc||Ps) increases with the ratio P c P s which in turn means that the reliability of steganalysis detectors will also increase. Accordingly, a stego technique is said to be perfectly secure if D(Pc||Ps) = 0 (Pc and Ps are equal), and

²-secure if the relative entropy between Pc and Ps is at most ², D(Pc||Ps) ≤ ². Perfectly secure algorithms are shown to exist, although they are impractical.

A type-I error, with probability α occurs, when a cover-object is mistaken for a stego-object (false alarm rate), and a type II error, with probability β, occurs when a stego-object is mistaken for a cover-object (miss rate). Thus bounds on these error probabilities can be computed using relative entropy, thereby relating steganographic security to detection error probabilities. Cachin obtains these bounds utilizing the facts that deterministic processing can not increase the relative entropy between two distributions, say, Pc and Ps, and hypothesis testing is a form of processing by a binary function that yields α (P(detect message present | message absent)) and β (P(detect message absent | message present)). Then, the relative entropy between distributions Pc and Ps and binary relative entropy of two distributions with parameters (α,1 − α) and (β, 1 − β) need to satisfy d(α, β) ≤ D(Pc||Ps), where d(α, β) is expressed as d(α, β) = α log α 1 − β + (1 − α) log 1 − α β . Then, for an ²-secure stego system we have d(α, β) ≤ ². (4) Consequently, when the false alarm rate is set to zero (α = 0), the miss rate is lower bound as β ≥ 2 −² . It should be noted that the probability of detection error for Wendy is defined as Pe = αP(message absent) + βP(message present). Based on above equations, for a perfectly secure stegosystem, α + β = 1, and when a cover-object is equally likely to undergo embedding operation, then Pe = 1 2

Hence, Wendy’s decisions are unreliable. As one can observe, there are several shortcomings in the above definition of security. While the ²-secure definition may work for random bit streams (with no inherent statistical structure), for real-life cover-objects such as audio, image, and video, it seems to fail. This is because real life cover-objects have a rich statistical structure in terms of correlation, higher-order dependence, etc.

# TECHNIQUES FOR IMAGE STEGANOGRAPHY

Given the proliferation of digital images, and given the high degree of redundancy present in a digital representation of an image (despite compression), there has been an increased interest in using digital images as cover-objects for the purpose of steganography. Therefore we have limited our discussion to the case of images for the rest of this tutorial. We should also note that there has been much more work on embedding techniques which make use of the transform domain or more specifically JPEG images due to their wide popularity. Thus to an attacker the fact that an image other than that of JPEG format is being transferred between two entities could hint of suspicious activity. There have been a number of image steganography algorithm proposed, these algorithm could be categorized in a number of ways:-

1. Spatial or Transform, depending on redundancies used from either domain for the embedding process.
2. Model based or ad-hoc, if the algorithm models statistical properties before embedding and preserves them, or otherwise.
3. Active or Passive Warden, based on whether the design of the embedder-detector pair takes into account the presence of an active attacker.

Another category for embedding techniques for which a number of algorithms have been proposed is the transform domain embedding category. Most of the work in this category has been concentrated on making use of redundancies in the DCT (discrete cosine transform) domain, which is used in JPEG compression. But there have been other algorithms which make use of other transform domains such as the frequency domain . Embedding in the DCT domain is simply done by altering the DCT coefficients, for example by changing the least significant bit of each coefficient. One of the constraints of embedding in the DCT domain is that many of the 64 coefficients are equal to zero, and changing two many zeros to non-zeros values will have an effect on the compression rate. That is why the number of bits one could embed in the DCT domain is less than the number of bits one could embed by the LSB method.

Also the embedding capacity becomes dependent on the image type used in the case of DCT embedding, since depending on the texture of image the number of non-zero DCT coefficients will vary. Although changing the DCT coefficients will cause unnoticeable visual artifacts, they do cause detectable statistical changes. As mentioned before, another transform domain which has been used for embedding is the frequency domain. Alturki et al. propose quantizing the coefficients in the frequency domain in order to embed messages. They first decorrelate the image by scrambling the pixels randomly, which in effect whitens the frequency domain of the image and increases the number of transform coefficients in the frequency domain thus increasing the embedding capacity. The frequency coefficients are then quantized to even or odd multiples of the quantization step size to embed zeros or ones. Then the inverse FFT of the signal is taken and descrambled. The resulting image would be visually incomparable to the original image. But statistically the image changes and as the authors show in their work, the result of the embedding operation is the addition of a gaussian noise to the image.

Nowadays and because of insecure networks and the internet that can be accessed by anyone it has become very risky to send important messages and files without any security measures. Before sending an important message we have to make sure that it will arrive at the destination in a secure way without being seen or modified by an intruder. Because of the threats mentioned before, steganography was discovered. Steganography in brief is the art of hiding information from other information. Usually we need to hide a text beneath any type of media. The most type of media used in hiding text is image. Steganography can be considered a new technology that still needs work and improvements.

Steganography is described by Neil F. Johnson and Sushil Jajodia in their paper Steganalysis: “The Investigation of Hidden Information” as The goal of Steganography is to avoid drawing suspicion to the transmission of a hidden message. If suspicion is raised, then this goal is defeated.‟ This technology is used widely nowadays, and a lot of people started hiding data, images, video and audio by using it. For this reason even if a message seems normal and innocent it might be holding within it a secret message with highly important data.

To clarify the idea of steganography, the three famous characters named Alice, Bob and Ward are used. Alice (A) wants to send a secret message (M) to Bob (B). Bob must receive it safely without raising suspicion. To do that, Alice changes the message (M) into a steganography object (stego-object, i.e. new file carrying the embedded-object) (S). Stego-object is created by covering the message (M) with another random harmless message to produce a cover (C, i.e. data file that will hold the secret message). Covering the massage

(M) with message (C) happens by using a secret key (stego-key) (K). Now Alice should be able to send the stego-object (S) to Bob without being detected by Ward. When Bob receives

(S) he will use the stego-key (K) which he already knows to reproduce secret message (M) from the cover message (C) and be able to read it. Steganography have to guarantee these requirements :

## 1. Robustness:

Information is robust when it is embedded inside an image and although it disappears behind it but it is not destroyed, it is present, but is only detected with reliability after modifying the image.

## Undetectability:



**2.**

The data hidden under an image cannot be detected as long as the cover image is not doubtable or suspicious and looks unchanged.

## Perceptual transparency:



**3.**

This requirement depends on the human visual and audio system. If the hidden data didn’t raise the attention of human systems and no one could distinguish whether the cover contains secret data then this requirement is guaranteed.

## Security:



**4.**

As long as no one other than the legal receiver can remove the embedded data from behind cover, the embedding algorithm is said to be secure. There may be different approaches for steganography methods classification. Our proposed technique belongs to the second classification type. Here are the six most used methods: Substitution, Injection (or insertion), Distortion, Generation, Transform domain techniques, Statistical steganography.

Internet technology provides many benefits for humans, especially in getting or exchanging information, learning, working, and others. One of the problems on the internet is security and data privacy. Many methods have been applied in providing security such as cryptography, steganography, watermarking, and digital signatures . Steganography and watermarking are branches of the science of hiding data on other media called cover . The difference is in its function, where steganography is used to hide messages while watermarking is used to preserve copyright. Data hiding in the image is divided into two domains, namely spatial domain and frequency domain. In the frequency domain, the message is hidden by first transforming the cover image. Transformations that have been widely used are DCT, DWT, and SVD .

This chapter applies LSB Steganography technique for various lossless file formats such as BMP, GIF and PNG. The science which deals with the hidden communication is called Steganography. There are different kinds of steganographic techniques which are complex and which have strong and weak points in hiding the invisible information in various file formats. The innocent carriers are the possible cover carriers which will hold the hidden communication.

A Steganography method is admirably secure only when the statistics of the cover information and the stego information are similar with each other. In other words it conveys the meaning that the relative entropy between the cover information and the stego information is zero. The LSB embedding technique suggests that data can be hidden in such a way that even the naked eye is unable to identify the hidden information in the LSBs of the cover file. In this chapter, a Steganography system is designed for hiding and unhiding a secret file into an image file using LSB insertion technique. An encryption and decryption technique on the data to be hidden into the image file is performed to provide additional security to the data. 47 Steganography is an alternative method for privacy and security. Instead of encrypting, we can hide the messages in another innocuous looking medium (carrier) so that their existence is not revealed.

# IMAGE STEGANOGRAPHY

Image compression is a technique which is widely used in Steganography. It is of two types- lossy compression and lossless compression. Lossy compression may not preserve the integrity of the original image whereas Lossless compression preserves the original image data correctly. Hence lossless compression is chosen. Examples of Lossless compression formats are GIF [84], BMP and PNG formats. JPEG format is the example for Lossy compression format.

# CHARACTERIZING DATA HIDING TECHNIQUES

An image is a picture that has been created or copied and stored in electronic form. An image can be described in terms of vector graphics or raster graphics . An image stored in raster form is sometimes called a bitmap . An image map is a file containing information that associates different locations on a specified image with hypertext links. An image is a collection of numbers that constitute different light intensities in different areas of the image.

This numeric representation forms a grid and the individual points are referred to as pixels (picture element).Greyscale images use 8 bits for each pixel and are able to display 256 different colours or shades of grey. Digital colour images are typically stored in 24-bit files and use the RGB colour model, also known as true colour. All colour variations for the pixels of a 24-bit image are derived from three primary colours: red, green and blue, and each primary colour is represented by 8 bits . Thus in one given pixel, there can be 256 different quantities of red, green and blue .

Steganography is a kind of technique which can embed a message inside a cover object. There are a number of features that characterize the merits and demerits of the embedding techniques. The way they are applied decides the importance of each and every feature. A set of criteria are proposed to define the invisibility of an algorithm. The criterias are as follows: Invisibility The imperceptibility of a Steganography technique is the most important necessity, since the quality of Steganography lies in its capacity to be unseen by the naked eye.

Payload Capacity Steganography techniques used aim at hiding the embedded secret data and also maximize the amount of information embedded. The amount of information that is hidden is called payload capacity. Hiding Capacity Concealing capacity is nothing but the size of data that could be concealed with respect to the size of the cover object. A vast concealing capacity permits the use of smaller cover images and thus decreases the data transmission needed to broadcast the stego image. Perceptual Transparency The inability of an eavesdropper to detect hidden data is referred to by Perceptual transparency.

# BINARY REPRESENTATION OF AN RGB COLOR IMAGE

For a 24-bit RGB image, every RGB component requires 8 bits of memory. The range of every RGB component value is in between 0 to 255 where 255 represents the brightest shade of the color and 0 represents darkest shade of the color. All different colors could be produced with the combination of these ranges. Subsequently, the test image is represented by an integer matrix. Every pixel is a mix of RGB values.

# ALGORITHMS USED IN PROPOSED METHOD

In the proposed method, Steganography is combined with Cryptography. It changes the meaning of the information as well as it hides the presence of information from the hacker. The LZ algorithm for compression and RSA algorithm for encryption and decryption are used in this chapter.

The easiest way to embed secret information within the cover file is called LSB insertion. If 24-bit color images are used, then the quantity of modification will be small. As an example, supposing that we have three neighbouring pixels (nine bytes) with the following RGB encoding:

50 01101010

11110010 00110110

01101001 11110000

00110101 01100000

11101111 00110100

Now if we wish to embed the following 9 bits of compressed secret information: 010010011. If we insert these 9 bits over the LSB of the 9 bytes above, we get the following sequence of bits (where bits in red color have been modified):

01101010 11110011

00110110 01101000

11110001 00110100

01100000 11101111

00110101

Note that we have successfully hidden 9 bits but at a cost of only modifying 5, or roughly 50% of the LSB bits. 3.4.2 LZ Compression Algorithm:

**Step-1:** Read the original file.

**Step-2:** Count the total number of words, alphabets, special characters and digits in the file.

**Step-3:** Find out the repeated words in the file.

**Step-4:** Prepare the word dictionary for the original file context.

**Step-5**: Create a compressed file. In the compressed file place the word's number instead of actual words.

**Step-6:** Add dictionary to compressed file.

**Step-7:** Save the compressed file along with the dictionary.

# RSA Algorithm

In Cryptography, RSA is an algorithm for public-key Cryptography. The RSA algorithm involves three steps: Key generation, encryption and decryption. Key Generation: The keys for the RSA algorithm are generated in the following way:

**Step-1:** Choose two different random prime numbers p and q.

**Step-2:** Compute n = p\*q. n is used as the modulus for both the private and public keys.

**Step-3:** Compute φ (n) = (p-1) (q-1). (φ is Euler’s totient function).

**Step-4:** Choose an integer e such that 1 < e < φ (pq), and gcd (e, φ(n))=1

**Step-5:** Compute d =e-1 mod [φ (n)]

**Step-6:** Publish the public encryption key: (e; n)

**Step-7:** Keep secret private decryption key: (d; n) 53 Encryption:

The steps required to encrypt information at sender are as follows:

**Step-1:** Obtain public key of recipient (e; n)

**Step-2:** Represent the information as an integer m in [0, n-1]

**Step-3:** Compute c = me mod n Decryption:

The steps required to decrypt information at receiver side are as follows:

**Step-1:** use private key (d; n)

**Step-2:** compute m = cd mod n

# IMPLEMENTATION OF LSB

IMPLEMENTATION OF LSB insertion is the easiest way to embed secret data in an image. By replacing the LSB of each sampling bit with a binary information, LSB insertion permits for a huge amount of secret information to be embedded. During the hiding and unhiding procedure, the content of the secret information should not be modified.

LSB Encoding Algorithm First the original image, and the compressed encrypted secret message are taken. Then the encrypted secret data has to be converted into binary format. Binary conversion is done by taking the American Standard Code of Information Interchange (ASCII) values of the character and converting them into binary format and generating a stream of bits. Same procedure is followed till all the message bits are placed in image bytes. Image generated is called ‘Stego-Image’. It is ready for transmission through the Internet. Algorithm for hiding secret data in Cover image:

**Step-1:** Read the cover image and secret text information which is to be embedded in to the cover image.

**Step-2:** Compress the secret information.

**Step-3:** Convert the compressed secret information into cipher text by using secret key shared by receiver and sender.

**Step-4:** Convert compressed encrypted text message into binary form.

**Step-5:** Find LSBs of each RGB pixel of the cover image.

**Step-6:** Embed the bits of the secret information into bits of LSB or RGB pixels of the cover image.

**Step-7:** Continue the procedure until the secret information is fully hidden into cover file.

The process is continued till the final count of the secret message bit is reached. After this, the bit stream of the message shall be generated. Available bits are grouped to form bytes such that each byte represents a single ASCII character. Characters are stored in a text file which represents the encrypted embedded message. After that the decryption and decompression are to be performed.

Algorithm for un hiding secret data from Stego image:

**Step-1:** Read the stego image.

**Step-2:** Find LSBs of each RGB pixel of the stego image.

**Step-3:** Find and retrieve the LSBs of each RGB pixel of the stego image.

**Step-4:** Continue the process until the message is fully extracted from the stego image. Step-5: Decompress the extracted secret data.

**Step-6:** Using shared key, decrypt secret information to get original information.

**Step-7:** Reconstruct the secret information.

Comparing cover image with stego image needs an image quality measure. Commonly used measures are MSE, PSNR, Correlation and Histogram . In statistics, MSE is quantifying the difference between values implied by an estimator and the true values of the quantity being estimated. MSE measures the average of the squares of the errors. PSNR scales the MSE according to the image range. The PSNR between cover image and stego image . A higher PSNR indicates that the quality of the stego image is similar to the cover image. Correlation, a best known method, not only evaluates the degree of closeness between two functions but also determines the extent to which the cover image and the stego image are close to each other even after embedding data.

According to Steganography, the secret message which is hidden may result in a distortionless image. At the same time this distortion will be perceptible to the naked eye. The quantity of information invisibly hidden in the image resulting in a distortionless image plays a pivotal role and this is decided by algorithm. The required characteristics are assessed while choosing a specific file format for Steganography. In the process of Steganography, the message which is hidden is invisible. An attempt has been made to implement encryption and decryption techniques on the data to be hidden into the carrier files, so that this will provide additional security to the data. The sender and receiver only know how to hide and 61 unhide the data into the carrier files.

To have the capacity to conceal secret information within a BMP image, one requires a substantial cover medium. The advantage of LSB hiding is its simplicity. LSB embedding technique also allows high perceptual transparency. The data hiding capacity of LSB techniques is high and more secure. Embedding secret information with Steganography techniques decreases the probability of secret information being detected. LSB insertion method to image Steganography works effectively for 24 BMP, GIF and PNG image file formats. Using these embedding and extracting algorithms, one can extract the secret message exactly as the original message without changing the cover image.

Hiding information into a medium requires following elements

* + - 1. The cover medium(C) that will hold the secret message.
      2. The secret message (M), may be plain text, digital image file or any type of data.
      3. The steganographic techniques
      4. A stego-key (K) may be used to hide and unhide the message.

In modern approach, depending on the cover medium, steganography can be divided into five types:

1. Text Steganography
2. Image Steganography
3. Audio Steganography
4. Video Steganography
5. Protocol Steganography
6. **Text steganography :** Hiding information in text files is the most common method of steganography. The method was to hide a secret message into a text message. After the coming of the Internet and different types of digital file formats it has decreased in importance. Text steganography using digital files is not used very often because the text files have a very small amount of excess data.
7. **Image steganography :** Images are used as the popular cover medium for steganography. A message is embedded in a digital image using an embedding algorithm, using the secret key. The resulting stego-image is sent to the receiver. On the other side, it is processed by the extraction algorithm Contents using the same key. During the transmission of stego- image unauthenticated persons can only notice the transmission of an image but cant see the existence of the hidden message.
8. **Audio steganography :** Audio steganography is concerned with embedding information in an innocuous cover speech in a secure and robust manner. Communication and transmission security and robustness are essential for transmitting vital information to intended sources while denying access to unauthorized persons. An audible sound can be inaudible in the presence of another louder audible sound .This property allows the user to select the channel in which to hide information . Existing audio steganography software can embed messages in WAV and MP3 sound files. The list of methods that are commonly used for audio steganography are listed and discussed below.
   * LSB coding
   * Parity coding
   * Phase coding
   * Spread spectrum
   * Echo hiding
   * Video steganography
9. **Video Steganography :** Video Steganography is a technique to hide any kind of files in any extension into a carrying Video file.
10. **Protocol steganography :** Protocol steganography The term protocol steganography is to embed information within network protocols such as TCP/IP. We hide information in the header of a TCP/IP packet in some fields that can be either optional or are never used.

# APPLICATIONS OF STEGANOGRAPHY

## Secret Communications

The use of steganography does not advertise secret communication and therefore avoids scrutiny of the sender, message, Contents and recipient. A trade secret, blueprint, or other sensitive information can be transmitted without alerting potential attackers.

## Feature Tagging Elements

Feature Tagging Elements can be embedded inside an image, such as the names of individuals in a photo or locations in a map. Copying the stego-image also copies all of the embedded features and only parties who possess the decoding stego-key will be able to extract and view the features.

## Copyright Protection

Copy protection mechanisms that prevent data, usually digital data, from being copied. The insertion and analysis of watermarks to protect copyrighted material is responsible for the recent rise of interest in digital steganography and data embedding.

The main advantage of the LSB coding method is a very high watermark channel bit rate; use of only one LSB of the host audio sample gives capacity of 44.1 kbps (sampling rate 44 kHz, all samples used for data hiding) and a low computational complexity. The obvious disadvantage is considerably low robustness, due to the fact that simple random changes of the LSBs destroy the coded watermark.

As the number of used LSBs during LSB coding increases or, equivalently, depth of the modified LSB layer becomes larger, probability of making the embedded message statistically detectable increases and perceptual transparency of stego objects is decreased. Therefore, there is a limit for the depth of the used LSB layer in each sample of host audio that can be used for data hiding.

None of the tested audio sequences had perceptual artifacts when the fourth LSB has been used for data hiding, although in certain music styles, the limit is even higher than the fourth LSB layer. Robustness of the watermark, embedded using the LSB coding method, increases with increase of the LSB depth used for data hiding. Therefore, improvement of watermark robustness obtained by increase of depth of the used LSB layer is limited by perceptual transparency bound, which is the fourth LSB layer for the standard LSB coding algorithm.

As the number of used LSBs during LSB coding increases or, equivalently, depth of the modified LSB layer becomes larger, probability of making the embedded message statistically detectable increases and perceptual transparency of stego objects is decreased. Therefore, there is a limit for the depth of the used LSB layer in each sample of host audio that can be used for data hiding. Subjective listening tests showed that, on average, the maximum LSB depth that can be used for LSB based watermarking without causing noticeable perceptual distortion is the fourth LSB layer when 16 bits per sample audio sequences are used. The tests were performed with a large collection of audio samples and individuals with different backgrounds and musical experience. None of the tested audio sequences had perceptual artifacts when the fourth LSB has been used for data hiding, although in certain music styles, the limit is even higher than the fourth LSB layer.

Robustness of the watermark, embedded using the LSB coding method, increases with increase of the LSB depth used for data hiding. Therefore, improvement of watermark robustness obtained by increase of depth of the used LSB layer is limited by perceptual transparency bound, which is the fourth LSB layer for the standard LSB coding algorithm. The main advantage of the LSB coding method is a very high watermark channel bit rate; use of only one LSB of the host audio sample gives capacity of 44.1 kbps (sampling rate 44 kHz, all samples used for data hiding) and a low computational complexity. The obvious disadvantage is considerably low robustness, due to the fact that simple random changes of the LSBs destroy the coded watermark.

As the number of used LSBs during LSB coding increases or, equivalently, depth of the modified LSB layer becomes larger, probability of making the embedded message statistically detectable increases and perceptual transparency of stego objects is decreased. Therefore, there is a limit for the depth of the used LSB layer in each sample of host audio that can be used for data hiding. Subjective listening tests showed that, on average, the maximum LSB depth that can be used for LSB based watermarking without causing noticeable perceptual distortion is the fourth LSB layer when 16 bits per sample audio sequences are used. The tests were performed with a large collection of audio samples and individuals with different backgrounds and musical experience. None of the tested audio sequences had perceptual artifacts when the fourth LSB has been used for data hiding, although in certain music styles, the limit is even higher than the fourth LSB layer.

Given the technology continues to evolve, more research is creating a hybrid technique of steganography and cryptography in order to provide layered security for messages . So this study combined steganographic techniques with LSB and cryptography by changing the contents of messages with XOR operators based on the three most significant bits. The scientific study in the open literature began in 1983 when Simmons stated the problem in terms of communication in a prison . In his formulation, two inmates Alice and Bob are trying to hatch an escape plan. The only way they can communicate with each other is through a public channel, which is carefully monitored by the warden of the prison ward. If Ward detects any encrypted messages or code, he will throw both Alice and Bob into solitary confinement. The problem of steganography is introduced then; how can Alice and Bob cook up an escape plan by communicating over the public channel in such a way that Ward doesn’t suspect “anything unusual” is going on. Notice how the goal of steganography is different from classical cryptography, which is about hiding the content of secret messages: steganography is about hiding the very existence of the secret message.

# LSB ENCODING

Least significant bit (LSB) coding is the simplest way to embed information in a digital audio file . By substituting the least significant bit of each sampling point with a binary message, LSB coding allows for a large amount of data to be encoded. The scientific study in the open literature began in 1983 when Simmons stated the problem in terms of communication in a prison . In his formulation, two inmates Alice and Bob are trying to hatch an escape plan. The only way they can communicate with each other is through a public channel, which is carefully monitored by the warden of the prison ward. If Ward detects any encrypted messages or code, he will throw both Alice and Bob into solitary confinement. The problem of steganography is introduced then; how can Alice and Bob cook up an escape plan by communicating over the public channel in such a way that Ward doesn’t suspect “anything unusual” is going on. Notice how the goal of steganography is different from classical cryptography, which is about hiding the content of secret messages: steganography is about hiding the very existence of the secret message .

Steganographic protocols have a long and intriguing history that goes back to antiquity. There are stories of secret messages written in invisible ink or hidden in love letters (the first character of each sentence can be used to spell a secret, for instance). More recently, steganography was used by prisoners and soldiers during World War II because all mail in Europe was carefully inspected at the time . Postal censors crossed out anything that looked like sensitive information (e.g. long strings of digits), and they prosecuted individuals whose mail seemed suspicious. In many cases, censors even randomly deleted innocent-looking sentences or entire paragraphs in order to prevent secret messages from going through. Over the last few years, steganography has been studied in the framework of computer science, and several algorithms have been developed to hide secret messages in innocent looking data .

**CHAPTER - 2**

# INTRODUCTION TO MATLAB

* 1. **INTRODUCTION**

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include

* + 1. Math and computation
    2. Algorithm development
    3. Data acquisition
    4. Modeling, simulation, and prototyping
    5. Data analysis, exploration, and visualization
    6. Scientific and engineering graphics

Application development, including graphical user interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non-interactive language such as C or FORTRAN. The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects.

Today, MATLAB engines incorporate the LAPACK and BLAS libraries, embedding the state of the art in software for matrix computation. MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science.

# THE MATLAB SYSTEM

The MATLAB system consists of five main parts:

## Development Environment:

This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and command window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path.

## The MATLAB Mathematical Function Library:

This is a vast collection of computational algorithms ranging from elementary functions, like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix Eigenvalues, Bessel functions, and fast Fourier transforms.

## The MATLAB Language:

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both

1. Programming in the smallǁ to rapidly create quick and dirty throw-away programs.
2. Programming in the largeǁ to create large and complex application programs.

## Graphics:

MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. The MATLAB Application Program Interface (API): This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

# 2.3. GRAPHICAL USER INTERFACE (GUI)

MATLAB’s Graphical User Interface Development Environment (GUIDE) provides a rich set of tools for incorporating graphical user interfaces (GUIs) in M-functions. Using GUIDE, the processes of laying out a GUI (i.e., its buttons, pop-up menus, etc.)and programming the operation of the GUI are divided conveniently into two easily managed and relatively independent tasks. The resulting graphical M-function is composed of two identically named (ignoring extensions) files:

A file with extension .m, called a GUI M-file, which contains the code that controls the GUI operation. This file includes functions that are called when the GUI is launched and exited, and callback functions that are executed when a user interacts with GUI objects for example, when a button is pushed. To launch GUIDE from the MATLAB command window, type guide filename Where filename is the name of an existing FIG-file on the current path. If filename is omitted, GUIDE opens a new (i.e., blank) window.

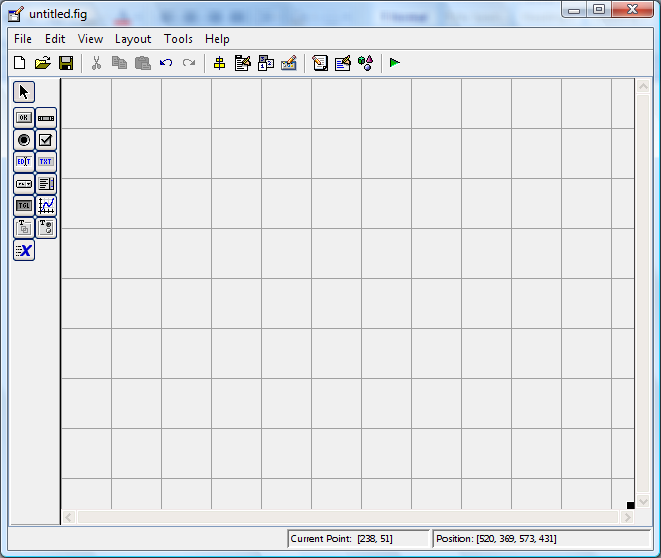


Fig 2.1 Matlab Procedure

A graphical user interface (GUI) is a graphical display in one or more windows containing controls, called components that enable a user to perform interactive tasks. The user of the GUI does not have to create a script or type commands at the command line to accomplish the tasks. Unlike coding programs to accomplish tasks, the user of a GUI need not understand the details of how the tasks are performed. GUI components can include menus, toolbars, push buttons, radio buttons, list boxes, and sliders just to name a few. GUIs created using MATLAB tools can also perform any type of computation, read and write data files, communicate with other GUIs, and display data as tables or as plots.

# GETTING STARTED

If you are new to MATLAB, you should start by reading Manipulating Matrices. The most important things to learn are how to enter matrices, how to use the: (colon) operator, and how to invoke functions. After you master the basics, you should read the rest of the sections below and run the demos. You will be rewarded with high productivity, high-creativity computing power that will change the way you work.

1. Introduction - describes the components of the MATLAB system.
2. Development Environment - introduces the MATLAB development environment, including information about tools and the MATLAB desktop.
3. Manipulating Matrices - introduces how to use MATLAB to generate matrices and perform mathematical operations on matrices.
4. Graphics - introduces MATLAB graphic capabilities, including information about plotting data, annotating graphs, and working with images.
5. Programming with MATLAB - describes how to use the MATLAB language to create scripts and functions, and manipulate data structures, such as cell arrays and multidimensional arrays.

# DEVELOPMENT ENVIRONMENT

* + 1. **Introduction**

This chapter provides a brief introduction to starting and quitting MATLAB, and the tools and functions that help you to work with MATLAB variables and files. For more information about the topics covered here, see the corresponding topics under Development Environment in the MATLAB documentation, which is available online as well as in print. Starting and Quitting MATLAB.

# Starting MATLAB

On a Microsoft Windows platform, to start MATLAB, double-click the MATLAB shortcut icon on your Windows desktop.On a UNIX platform, to start MATLAB, type matlab at the operating system prompt. After starting MATLAB, the MATLAB desktop opens - see MATLAB Desktop. You can change the directory in which MATLAB starts, define startup options including running a script upon startup, and reduce startup time in some situations.

# Quitting MATLAB

To end your MATLAB session, select Exit MATLAB from the File menu in the desktop, or type quit in the Command Window. To execute specified functions each time MATLAB quits, such as saving the workspace, you can create and run a finish.m script.

# MATLAB Desktop

When you start MATLAB, the MATLAB desktop appears, containing tools (graphical user interfaces) for managing files, variables, and applications associated with MATLAB. The first time MATLAB starts, the desktop appears as shown in the following illustration, although your Launch Pad may contain different entries. You can change the way your desktop looks by opening, closing, moving, and resizing the tools in it.

You can also move tools outside of the desktop or return them back inside the desktop (docking). All the desktop tools provide common features such as context menus and keyboard shortcuts. You can specify certain characteristics for the desktop tools by selecting Preferences from the File menu. For example, you can specify the font characteristics for Command Window text. For more information, click the Help button in the Preferences dialog box.

# Desktop Tools

This section provides an introduction to MATLAB's desktop tools. You can also use MATLAB functions to perform most of the features found in the desktop tools. The tools are:

Current Directory Browser Workspace Browser Array Editor Editor/Debugger Command Window Command History Launch Pad

Help Browser

Command Window Use the Command Window to enter variables and run functions and M-files. Command History Lines you enter in the Command Window are logged in the Command History window. In the Command History, you can view previously used functions, and copy and execute selected lines. To save the input and output from a MATLAB session to a file, use the diary function.

## Running External Programs :

You can run external programs from the MATLAB Command Window. The exclamation point character! is a shell escape and indicates that the rest of the input line is a command to the operating system. This is useful for invoking utilities or running other

programs without quitting MATLAB. On Linux, for example,!emacsmagik.m invokes an editor called emacs for a file named magik.m. When you quit the external program, the operating system returns control to MATLAB.

## Launch Pad:

MATLAB's Launch Pad provides easy access to tools, demos, and documentation.

## Help Browser:

Use the Help browser to search and view documentation for all your Math Works products. The Help browser is a Web browser integrated into the MATLAB desktop that displays HTML documents.The Help browser consists of two panes, the Help Navigator, which you use to find information, and the display pane, where you view the information.

## Help Navigator:

Use Help Navigator to find information. It includes:

1. Product filter - Set the filter to show documentation only for the products you specify.
2. Contents tab - View the titles and tables of contents of documentation for your products.
3. Index tab - Find specific index entries (selected keywords) in the MathWorks documentation for your products.
4. Search tab - Look for a specific phrase in the documentation. To get help for a specific function, set the Search type to Function Name.
5. Favorites tab - View a list of documents you previously designated as favorites.

After finding documentation using the Help Navigator, view it in the display pane. While viewing the documentation, you can:

Browse to other pages - Use the arrows at the tops and bottoms of the pages, or use the back and forward buttons in the toolbar.

Bookmark pages - Click the Add to Favorites button in the toolbar. Print pages - Click the print button in the toolbar.

Find a term in the page - Type a term in the Find in page field in the toolbar and click Go. Other features available in the display pane are: copying information, evaluating a selection, and viewing Web pages.

## Current Directory Browser:

MATLAB file operations use the current directory and the search path as reference points. Any file you want to run must either be in the current directory or on the search path.

## Search Path:

To determine how to execute functions you call, MATLAB uses a search path to find M-files and other MATLAB-related files, which are organized in directories on your file system. Any file you want to run in MATLAB must reside in the current directory or in a directory that is on the search path. By default, the files supplied with MATLAB and MathWorks toolboxes are included in the search path.

## Workspace Browser

The MATLAB workspace consists of the set of variables (named arrays) built up during a MATLAB session and stored in memory. You add variables to the workspace by using functions, running M-files, and loading saved workspaces. To view the workspace and information about each variable, use the Workspace browser, or use the functions who and whos. To delete variables from the workspace, select the variable and select Delete from the Edit menu. Alternatively, use the clear function. The workspace is not maintained after you end the MATLAB session.

## Array Editor

Double-click on a variable in the Workspace browser to see it in the Array Editor. Use the Array Editor to view and edit a visual representation of one- or two-dimensional numeric arrays, strings, and cell arrays of strings that are in the workspace.

## Editor/Debugger

Use the Editor/Debugger to create and debug M-files, which are programs you write to run MATLAB functions. The Editor/Debugger provides a graphical user interface for basic text editing, as well as for M-file debugging. You can use any text editor to create M-files, such as Emacs, and can use preferences (accessible from the desktop File menu) to specify that editor as the default. If you use another editor, you can still use the MATLAB Editor/Debugger for debugging, or you can use debugging functions, such as dbstop, which sets a breakpoint.

**CHAPTER-3**

**IMAGE STEGANOGRAPHY TECHNIQUES**

There are several Steganographic techniques for image file format which are as follows:

# SPATIAL DOMAIN TECHNIQUE

There are many versions of spatial steganography, all directly changing some bits in the image pixel values in hiding data.

**Least significant bit (LSB)-**based steganography is one of the simplest techniques that hides a secret message in the LSBs of pixel values without perceptible distortions. To our human eye, changes in the value of the LSB are imperceptible. Embedding of message bits can be done either simply or randomly. Least Significant Bit (LSB) replacement technique, Matrix embedding, are some of the spatial domain techniques. Advantages of spatial domain LSB technique are:

1. Degradation of the original image is not easy.
2. Hiding capacity is more i.e. more information can be stored in an image.

Disadvantages of LSB technique are:

* 1. Robustness is low
  2. Hidden data can be destroyed by simple attacks.

# MASKING AND FILTERING

It is a steganography technique which can be used on gray-scale images. Masking and filtering is similar to placing watermarks on a printed image. These techniques embed the information in the more significant areas than just hiding it into the noise level. Watermarking techniques can be applied without the fear of image destruction due to lossy compression as

they are more integrated into the image. Advantages of Masking and filtering Techniques: This method is much more robust than LSB replacement with respect to compression. Disadvantages: Techniques can be applied only to gray scale images and restricted to 24 bits.

# TRANSFORM AND DOMAIN TECHNIQUE

The Frequency domain the message is inserted into transformed coefficients of image giving more information hiding capacity and more robustness against attacks. Transform domain embedding can be termed as a domain of embedding techniques for which a number of algorithms have been suggested. Most of the strong steganographic systems today operate within the transform domain. Transform domain techniques have an advantage over LSB techniques as they hide information in areas of the image that are less exposed to compression, cropping, and image processing.

Some transform domain techniques do not seem dependent on the image format and they may outrun lossless and lossy format conversions. Transform domain techniques are of different type[3]:

* + 1. Discrete Fourier transformation technique (DFT).
    2. Discrete cosine transformation technique (DCT).
    3. Discrete Wavelet transformation technique (DWT).

# DISTORTION TECHNIQUES

In this technique, store information by signal distortion and measure the deviation from the original cover in the decoding process. Distortion techniques need knowledge of the original cover image during the decoding process where the decoder functions to check for differences between the original cover image and the distorted cover image in order to restore the secret message. In this technique, a stego-image is created by applying a sequence of modifications to the cover image.

This sequence of modifications is used to match the secret message required to transmit. The message is encoded at pseudo-randomly chosen pixels. If the stego image is different from the cover image at the given message pixel, the message bit is a 1. Otherwise, the message bit is a 0. The encoder can modify the 1 value pixels in such a manner that the statistical properties of the image are not affected. If an attacker interferes with the stego-image by cropping, scaling or rotating, the receiver can easily detect it.

# CHARACTERISTICS FEATURE OF DATA HIDING TECHNIQUES

Perceptibility shows embed message distort cover medium to a visually unacceptable level. Capacity shows How much information can be hidden relative to the change in perceptibility.

Robustness Attacks can include manipulation of the stego medium in an effort to destroy, or change the embedded data.

Tamper Resistance Beyond robustness to destruction, tamper-resistance refers to the difficulty for an attacker to alter a message once it has been embedded in a stego-image.

# IMAGE STEGANALYSIS

Steganalysis is the breaking of steganography and is the science of detecting hidden information . The main objective of steganalysis is to break steganography and the detection of stego image. Almost all steganalysis algorithms depend on steganographic algorithms introducing statistical differences between cover and stego image. Steganalysis are of three different types:

Visual attacks discovered the hidden information, which helps to separate the image into bit planes for furthermore analysis.

# STATISTICAL ATTACK

Statistical attacks may be passive or active. Passive attacks involve identifying the presence or absence of a secret message or embedding algorithm used. Active attacks are used to investigate embedded message length or hidden message location or secret key used in embedding. Structural attacks The format of the data files changes as the data to be hidden is embedded, identifying this characteristic structure changes can help us to find the presence of image/text file.

# STEGANALYSIS TOOLS

There are several steganalysis tools available in the market like PhotoTitle, 2Mosaic and StirMark Benchmark etc. These three steganalytic tools can remove steganographic content from any image. This is achieved by destroying secret messages by two techniques: break apart and resample.

# LEAST- SIGNIFICANT BIT(LSB) TECHNIQUES

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. Digital images are mainly of two types

1. 24 bit images and
2. 8 bit images.

In 24 bit images we can embed three bits of information in each pixel, one in each LSB position of the three eight bit values. Increasing or decreasing the value by changing the LSB does not change the appearance of the image; much so the resultant stego image looks almost the same as the cover image.

# COVER AND HIDDEN IMAGES

The hidden image is extracted from the stego-image by applying the reverse process. If the LSB of the pixel value of cover image C(i,j) is equal to the message bit m of the secret message to be embedded, C(i,j) remains unchanged; if not, set the LSB of C(i, j) to m. The message embedding procedure is given below

S(i,j) = C(i,j) - 1, if LSB(C(i,j)) = 1 and m = 0

S(i.j) = C(i,j), if LSB(C(i,j)) = m

S(i,j) = C(i,j) + 1, if LSB(C(i,j)) = 0 and m = 1 where LSB(C(i, j)) stands for the LSB of cover image C(i,j) and m is the next message bit to be embedded. S(i,j) is the stego image As we already know each pixel is made up of three bytes consisting of either a 1 or a 0. For example, suppose one can hide a message in three pixels of an image (24-bit colors). Suppose the original 3 pixels are:

(11101010 11101000 11001011)

(01100110 11001010 11101000)

(11001001 00100101 11101001)

A steganographic program could hide the letter ”J” which has a position 74 into ASCII character set and have a binary representation ”01001010”, by altering the channel bits of pixels.

(11101010 11101001 11001010)

(01100110 11001011 11101000)

(11001001 00100100 11101001)

In this case, only four bits needed to be changed to insert the character successfully. The advantage of LSB embedding is its simplicity and many techniques use these methods . LSB embedding also allows high perceptual transparency.

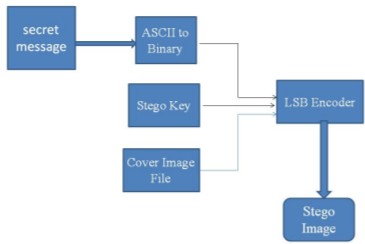


Fig 3.1 LSB insertion technique

# DATA EMBEDDING

The embedding process is as follows:

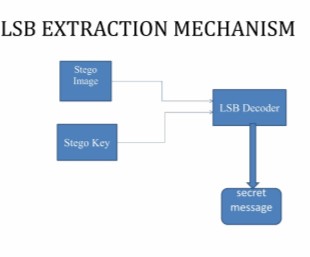
Inputs Cover image, stego-key and the text file, Output stego image

Fig 3.2 LSB extraction mechanism

## Procedure :

**Step 1:** Extract the pixels of the cover image. **Step 2:** Extract the characters of the text file. **Step 3:** Extract the characters from the Stego key.

**Step 4:** Choose the first pixel and pick characters of the Stego key and place it in the first component of the pixel.

**Step 5:** Place some terminating symbol to indicate the end of the key. 0 has been used as a terminating symbol in this algorithm.

**Step 6:** Insert characters of the text file in each first component of next pixels by replacing it.

**Step 7:** Repeat step 6 till all the characters have been embedded.

**Step 8:** Again place some terminating symbol to indicate the end of data.

**Step 9:** Obtained stego image.

# DATA EXTRACTION

The extraction process is as follows. Inputs Stego-image file, stego-key Output Secret text message.

## Procedure :

**Step 1:** Extract the pixels of the stego image.

**Step 2:** Now, start from the first pixel and extract stego key characters from the first component of the pixels. Follow

**Step3:** Up to terminating symbol, otherwise follow step 4.

**Step 4:** If this extracted key matches with the key entered by the receiver, then

follow Step 5, otherwise terminate the program.

**Step 5:** If the key is correct, then go to the next pixels and extract secret message characters from the first component of the next pixels. Follow Step 5 till up to the terminating symbol, otherwise follow step 6.

**Step 6:** Extract secret message.

# IMAGE ENCODING ALGORITHM

Inputs Image file, stego key and image file Output Stego image.

* + 1. The cover and secret images are read and converted into the uint8 type.
    2. The numbers in the secret image matrix are conveyed to 8-bit binary. Then the matrix is reshaped to a new matrix a.
    3. The matrix of the cover image is also reshaped to matrix b
    4. Perform the LSB technique described above
    5. The stego-image, which is very similar to the original cover image, is achieved by reshaping matrix b.
    6. While extracting the data, the LSB of the stego image is collected and they are reconstructed into the decimal numbers. The decimal numbers are reshaped to the secret image.

# PSEUDO-RANDOM ENCODING TECHNIQUE

In this technique, A random key is used to choose the pixels randomly and embed the message. This will make the message bits more difficult to find and hopefully reduce the realization of patterns. Data can be hidden in the LSB of a particular colour plane (Red plane) of the randomly selected pixel in the RGB colour space.

# EMBEDDING ALGORITHM

In this process of encoding method, a random key is used to randomise the cover image and then hide the bits of a secret message into the least significant bit of the pixels within a cover image. The transmitting and receiving end share the stego key and random-key. The random-key is usually used to seed a pseudo-random number generator to select pixel locations in an image for embedding the secret message.

Inputs Cover image, stego-key and the message Output stego image

* + - 1. Read character from text file that is to be hidden and convert the ASCII value of the character into equivalent binary value into an 8 bit integer array.
      2. Read the RGB colour image(cover image) into which the message is to be embedded.
      3. Read the last bit of red pixel.
      4. Initialize the random key and Randomly permute the pixels of cover image and reshape into a matrix.
      5. Initialize the stego-key and XOR with a text file to be hidden and give a message.
      6. Insert the bits of the secret message to the LSB of the Red plane’s pixels.
      7. Write the above pixel to the Stego Image File.

# EXTRACTION OF HIDDEN MESSAGE

In this process of extraction, the process first takes the key and then random-key. These keys take out the points of the LSB where the secret message is randomly distributed . Decoding process searches the hidden bits of a secret message into the least significant bit of the pixels within a cover image using the random key. In the decoding algorithm the random-key must match i.e. the random-key which was used in encoding should match because the random key sets the hiding points of the message in case of encoding. Then the receiver can extract the embedded messages exactly using only the stego-key.

# MESSAGE EXTRACTION ALGORITHM

Inputs Stego-image file, stego-key,random key. Output Secret message.

* + - 1. Open the Stego image file in read mode and from the Image file, read the RGB colour of each pixel.
      2. Extract the red component of the host image.
      3. Read the last bit of each pixel.
      4. Initialize the random-key that gives the position of the message bits in the red pixel that are embedded randomly.
      5. For decoding, select the pixels and Extract the LSB value of red pixels.
      6. Read each of the pixels then the content of the array converts into a decimal value that is actually the ASCII value of the hidden character.
      7. ASCII values are XOR with a stego-key and gives a message file, which we hide inside the cover image.

# GRAY SCALE

The following chart displays all 256 Gray-scale colours. The gray-scale colour naming scheme uses a two digit hex value to define up to 256 shades of gray. In photography and computing, a grayscale digital image is an image in which the value of each pixel is a single sample, that is, it carries only intensity information. Images of this sort, also known as black-and-white, are composed exclusively of shades of gray, varying from black at the weakest intensity to white at the strongest.

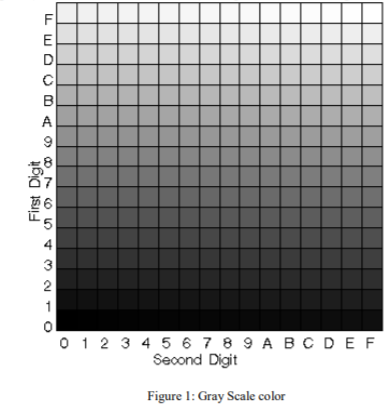
Gray scale images are distinct from one-bit bi-tonal black-and-white images, which in the context of computer imaging are images with only the two colours, black (also called bi-level or binary images). Gray scale images have many shades of gray in between. Gray scale images are often the result of measuring the intensity of light at each pixel in a single band of the electromagnetic spectrum (e.g. infrared, visible light, ultraviolet, etc.), and in such cases they are monochromatic when only a given frequency is captured. Grayscale Shading Strengths (0=no colour; 15=full colour).

Fig 3.3. Gray Scale Color

A message is embedded into the image by the stego system encoder via a secret key or password. This password or secret key should be kept secret. The resulting stego image is transmitted over a channel to the receiver. The stego system at the decoder end, using the same key or password, will decode the stego image.

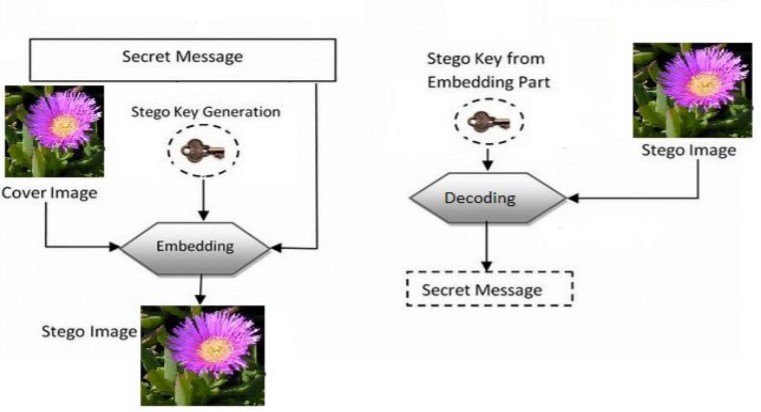


Fig 3.4 Steganography Block Diagram

There are two important components, cover image and hiding data, in data hiding technique. The cover image I is an 8-bit gray scale image. The size of the cover image is m×n. The hiding data H embedded in I is g-bits bit stream. We use the equation below to express image C, data D and each pixel separately.I= { } H= One of the simplest systems for embedding digital data into a digital cover is the Least Significant Bit method.

Consider an N×M image in which each pixel value is represented by a decimal number in the range determined by the number of bits used. In a gray-scale image, with 8 bit precision per pixel, each pixel assumes a value between [0, 255] and each positive number P can be represented by: P= + = This property allows the decomposition of an image into a collection of binary images by separating them into n bit planes.

In the classical LSB embedding methods, the secret message is inserted into the least-significant bit plane of the cover image either by directly replacing those bits. The amount of data to be embedded may also be fixed or variable in size depending on the number of pixels selected. The main advantage of such a technique is that the modification of the LSB plane does not affect the human perception of the overall image quality as the amplitude variation of the pixel values is bounded by ±1. The masking properties of the Human Visual System allow significant amounts of embedded information to be unnoticed by imperceptible by the average observer under normal viewing conditions. “Masking” refers to the phenomenon where a signal can be imperceptible to an observer in the presence of another signal. A detailed review of these techniques is given in. Other advantages of LSB data hiding included high embedding capacity and low computational complexity. The main disadvantages are the weaknesses with respect to robustness, tampering, geometric attacks, filtering, and compression.

In LSB steganography, the least significant bits of the cover media’s digital data are used to conceal the message. The simplest of the LSB steganography techniques is LSB replacement. LSB replacement steganography changes the last bit of each of the pixel values to reflect the message that needs to be hidden. Consider an 8- bit grayscale bitmap image where each pixel is stored as a byte representing a gray scale color value.

Suppose the first eight pixels of the original image have the following gray color values: 01010010

01001010

10010111

11001100

11010101

01010111

00100110

01000011

To hide the letter Z whose binary value of ASCII [11] code is 10110101, we would replace the LSBs of these pixels to have the following new values:

01010011

01001010

10010111

11001101

11010100

01010111

00100110

01000011

Note that, on average, only half the LSBs need to change. The difference between the cover (i.e. original) image and the stego image will be hardly noticeable to the human eye. However, one of its major limitations is the small size of data which can be embedded in such types of images using only LSB. LSB is extremely vulnerable to attacks. LSB techniques implemented to 24 bit formats for the color image are difficult to detect contrary to 8 bit format.

# LEAST SIGNIFICANT BIT TECHNIQUE

Least significant bit (LSB) insertion is a common and simple approach to embed information in an image file. In this method the LSB of a byte is replaced with an M‟ s bit. This technique works well for image steganography. To the human eye the stego image will look identical to the carrier image. For hiding information inside the images, the LSB (Least Significant Byte) method is usually used. To a computer an image file is simply a file that shows different colors and intensities of light on different areas of an image.

The best type of image file to hide information inside is a 24 Bit BMP (Bitmap) image. When an image is of high quality and resolution it is easier to hide information inside the image. Although 24 Bit images are best for hiding information due to their size The least significant bit i.e. the eighth bit is used to change to a bit of the secret message. When using a 24-bit image, one can store 3 bits in each pixel by changing a bit of each of the red, green and blue color components.

Suppose that we have three adjacent pixels (9 bytes) with the RGB encoding: 10010101

00001101

11001001

10010110

00001111

11001011

10011111

00010000

11001011

When the number 300, can be which binary representation is 100101100 embedded into the least significant bits of this part of the image. If we overlay these 9 bits over the LSB of the 9 bytes above we get the following (where bits in bold have been changed) 10010101

00001100 11001000

10010111 00001110

11001011 10011111

00010000 11001010

Here the number 300 was embedded into the grid, only the 5 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.

* + 1. Select a cover image of size M\*N as an input.
    2. The message to be hidden is embedded in the RGB component only of an image.
    3. Use a pixel selection filter to obtain the best areas to hide information in the cover image to obtain a better rate. The filter is applied to Least Significant Bit (LSB) of every pixel to hide information, leaving the most significant bits (MSB).
    4. After that Message is hidden using the Bit Replacement method.

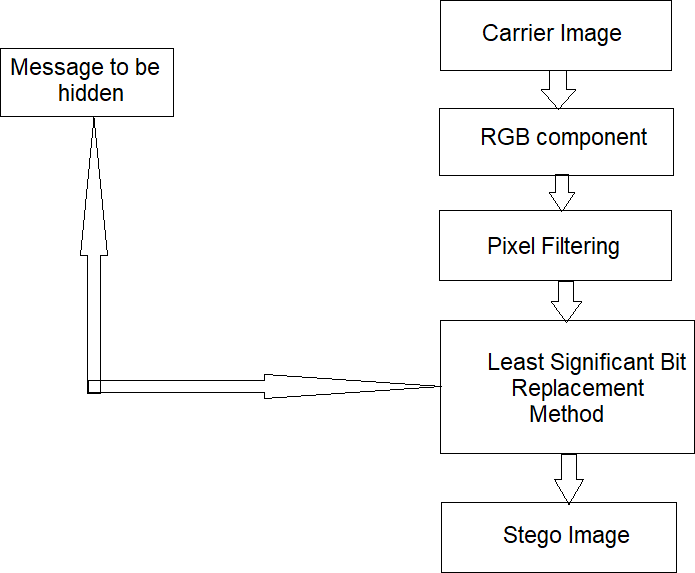


Fig 3.5 Algorithm of Least Significant Bit

**CHAPTER - 4**

**RELATED WORK**

K. Joshi, et.al., drew up steganographic methods on spatial domains using the XOR operator. Message insertion is performed with two first XOR operations imposed on the 1st and 8th bits and the second is on the 2nd, 7th bit. The result of the operation is then compared and used as a rule for inserting the message. The cover image used is a 512 \* 512 grayscale image and three message sizes, 1024 bits, 2048 bits and 4096 bits. The value of PSNR obtained is about 69 dB with a message length of 4096 bits.

A. U. Islam, et.al., proposing steganography on the image using the differencing bits technique on the 5th and 6th bits. If there is a difference in the 5th and 6th bits equal to the bits of the secret information then no change is made. Meanwhile, if there is a difference in value, the value changes to the 5th bit so that the value of the difference corresponds to the bit value in the secret information. The cover image used is grayscale image and color image with size 512 \* 512. With this method obtained PSNR 51.17977 dB on Lena grayscale image and PSNR 52.3438 on Lena color image, with a payload capacity of 262144 bits message.

C. Irawan, et.al. proposed a combination of steganographic and cryptographic methods, with messages encrypted using the OTP method before being embedded on the LSB. To improve imperceptibility and secure embedding of messages is done on the image edge area. Image edge area detection is done by the Canny method. The cover image used is type JPEG with the size of 11035 bytes, while the message pinned 1024 bytes obtained PSNR 69.1106 dB. In addition, the stego image quality is also measured by a histogram, where the cover image histogram and stego image are identical.

E. J. Kusuma, et. al also proposes embedding messages in the image edge area. In his research combined steganography techniques using LSB and cryptographic techniques using DES. Before the image message is inserted, the message is encrypted using the DES method. The cover image used is a color image with the size 1024 \* 1024, while the message is also a color image with size 64 \* 64. The cover image used is type JPEG with the size of 11035 bytes, while the message pinned 1024 bytes obtained PSNR 69.1106 dB.

The encrypted message is embedded in the image edge area detected by the Canny method. Based on the results of this research, the average value of PSNR is 72.21584 dB, which is obtained from five kinds of imagery.

K. Joshi and R. Yadav , proposed steganographic techniques with LSBs combined with XOR operators. The message is pinned on the smallest three bits. Where before being embedded secret messages carried XOR operation. In the grayscale cover image size, 256 \* 256 pixels can be embedded with a maximum 196608 bits message. While the PSNR value obtained at the maximum embed is above 37 dB.

**CHAPTER -5**

**DESIGN AND IMPLEMENTATION**

In the method we propose, there are two main schemes, namely message embedding and message extraction. For more details can be seen in the sub-section below.

# Embedding Scheme

In the embedding, the scheme required input in the form of gray-scale cover image and image of the message in the form of a binary image with the exact same size. While the output obtained is a stego image. For more details can see the visualization in Figure below

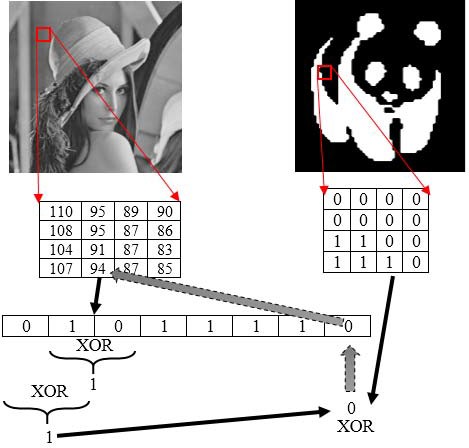


Fig 5.1 Embedded Scheme

Here are the steps for the embedding process:

Step 1: Read the cover image (A) and message image (B). Step 2: Change the pixel value to binary.

Step 3: Perform XOR operations on the 7th and on the 6th bit.

Step 4: Perform XOR operation on bit 8th with XOR operation result on the 7th & 6th bit. Step 5: Perform XOR operations on message bits with three MSB bits (8th, 7th, and 6th bits).

Step 6: Save the XOR operation result in the message bit, then convert again to uint8, the result of this conversion being the stego image pixel value.

# Extraction Scheme

The extraction scheme only required input in the form of stego image. While the output of the recovered message in the form of a binary image.

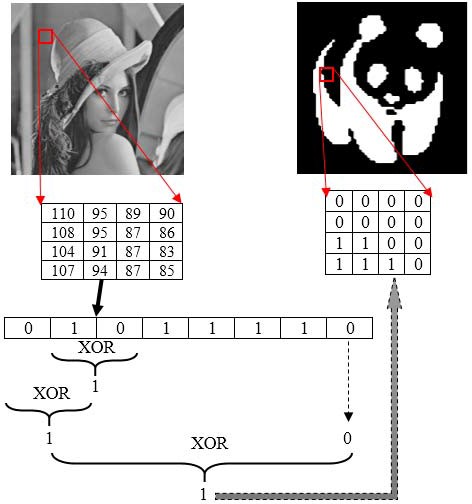


Fig 5.2 Extraction Scheme

Figure 2 shows the visualization of the extraction process, in detail, the steps in the extraction process are as follows:

Step 1: Read the stego image (S).

Step 2: Change the pixel value to binary.

Step 3: Perform XOR operations on the 7th and 6th bits.

Step 4: Perform XOR operation on the 8th bit with XOR operation result on the 7th and on the 6th bit.

Step 5: Do the XOR operation on the LSB with three bits MSB (bits 8th, 7th, and 6th).

Step 6: Save and collect the results of the XOR operation on the LSB, then convert again to uint8, the result of this conversion being a recovery of the message image.

**CHAPTER-6**

**EXPERIMENTS AND RESULTS**

Experiments conducted in this research using six grayscale images as the cover image, and binary image as a message, as cover images and Figure 4 as a message. The cover image and message image have the same size, 256 \* 256 in size. Here is a cover image used in this study:

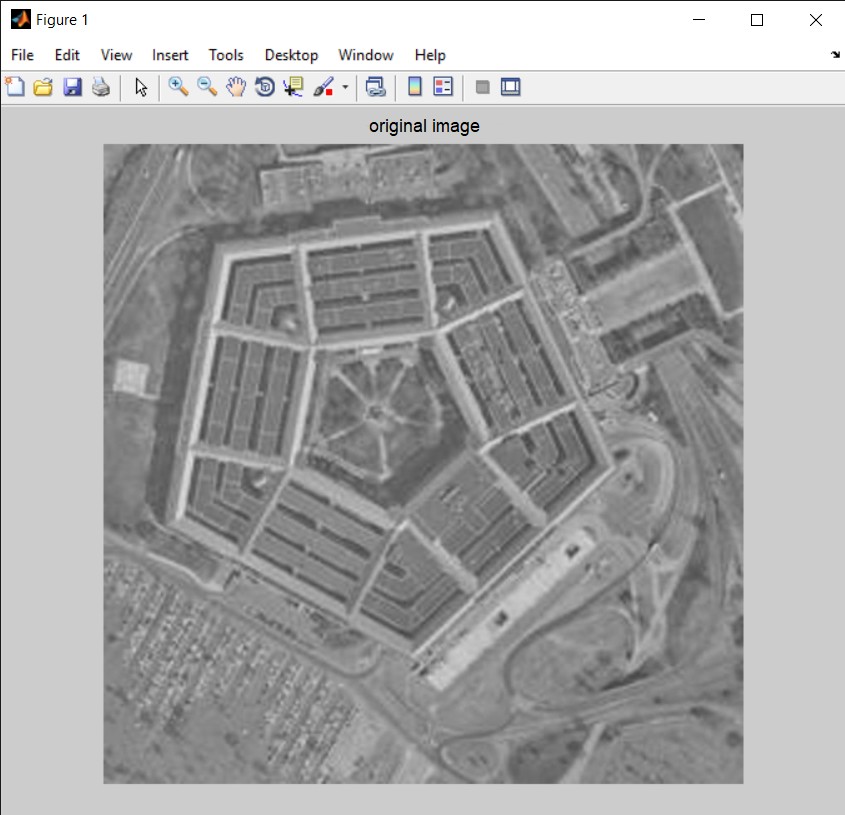


FIG 6.1 Original Image

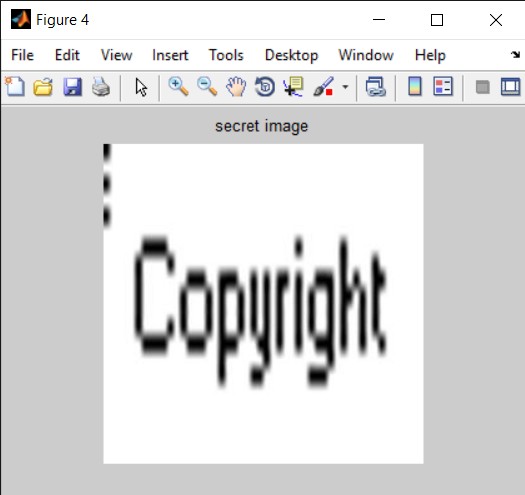


Fig. 6.2 Message Image Used

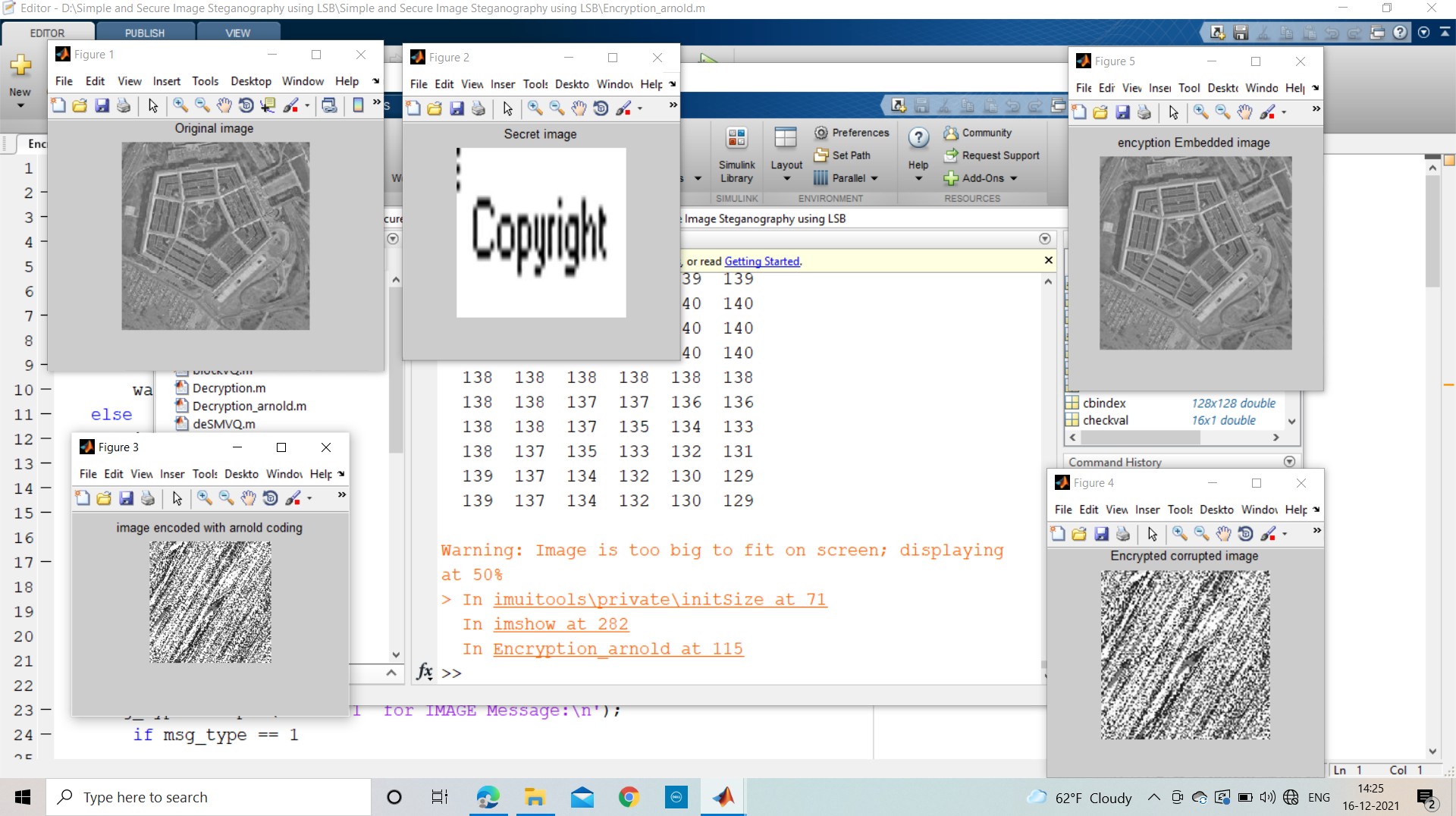


Fig 6.3 Output of Encryption

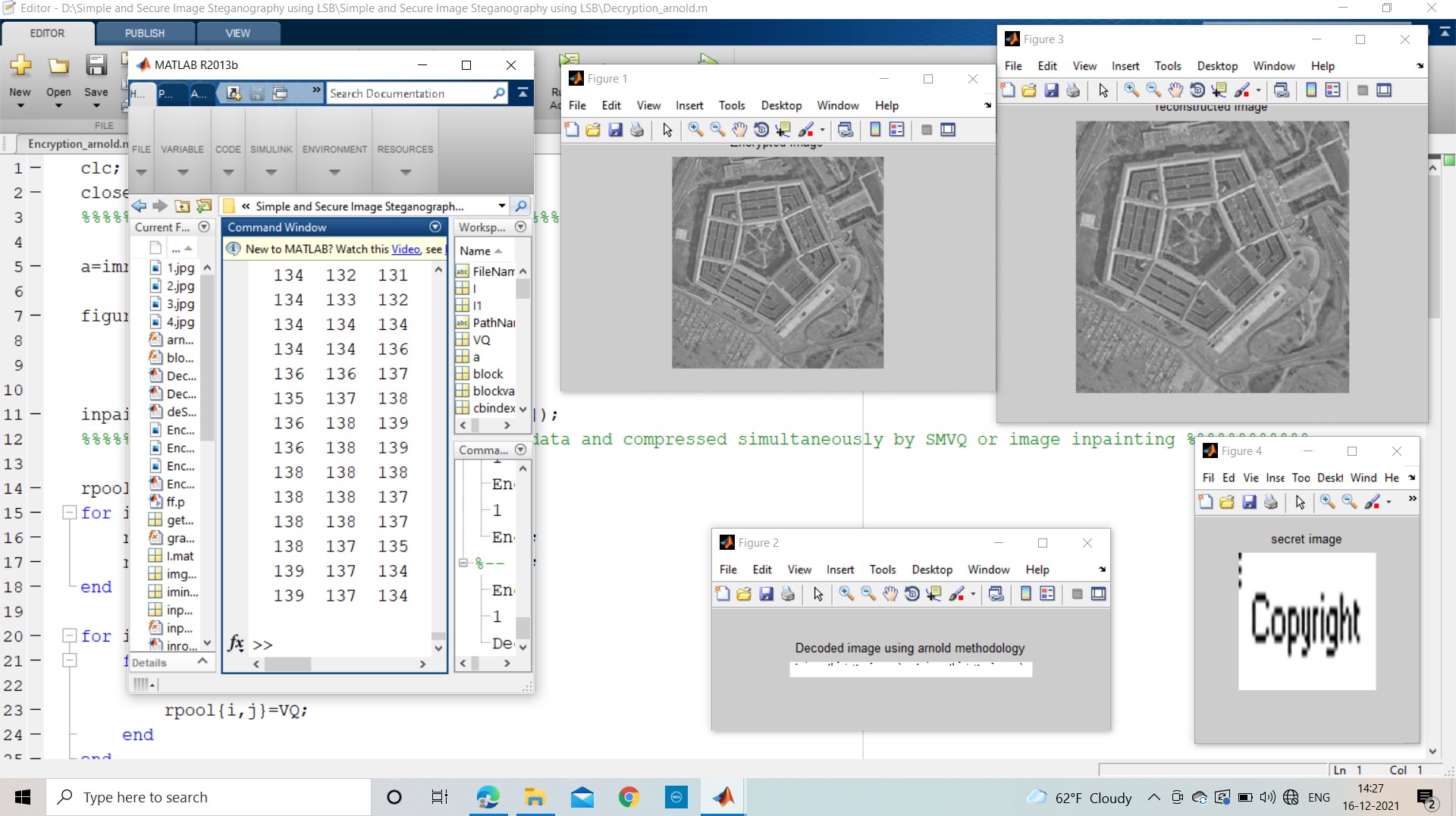


Fig. 6.4 Output of Decryption

The cover image and the message image have the same pixel size. Then one pixel of the message image embedded on one cover image pixel. The number of message bits embedded in the cover image is 65536 bits. Furthermore, an experiment of embedding the message using the method proposed above.

**CHAPTER-7**

**CONCLUSION**

The method proposed in this study has an advantage in the aspect of imperceptibility as evidenced by the excellent value of PSNR and MSE. Where all PSNR values are more than 50dB, so does the MSE value not more than 0.3. This method is also very simple and safe because with XOR operation the steganography process can be done quickly and easily. With the XOR operator, the embedded bits cannot be directly guessed. Moreover, there are three keys used, with three times the XOR operation. The use of an integrated key in the cover image also keeps the stego file the same size, and no key delivery is required to the receiver so it can speed up the messaging process as the file size is maintained. However, based on histogram analysis there is a distinct pattern difference between the cover image and stego image.

**FUTURE SCOPE**

The previous segment gives subtleties of different sorts of Techniques of that have developed after some time as for the idea of the cover picture and the separate areas. Aside from this, the first area likewise proposes a few qualities that are very essential for a decent steganography framework. Fusing these into solitary framework is itself only broad research.

Security and Capacity exchange off is an imperative issue in steganography. It has been seen that expans-

-ion in the limit prompts giving up the security to some degree. There has not been much hypothetical

investigation in relating the security and limit parameters scientifically.

**CHAPTER - 8**

**REFERENCES**

1. W. S. Sari, E. H. Rachmawanto, D. R. I. M. Setiadi and h. A. Sari, "A Good Performance OTP Encryption Image based on DCT-DWT Steganography," TELKOMNIKA (Telecommunication Computing Electronics and Control) , vol. 15, no. 4, pp. 1987-1995, 2017.
2. R. D. Ardy, O. R. Indriani, C. A. Sari, D. R. I. M. Setiadi and E. H. Rachmawanto, "Digital Image Signature using Triple Protection Cryptosystem (RSA, Vigenere, and MD5)," in International Conference on Smart Cities, Automation & Intelligent Computing Systems (ICONSONICS), Yogyakarta, 2017.
3. A. Winarno, D. R. I. M. Setiadi, A. A. Arrasyid, C. A. Sari and E. H. Rachmawanto, "Image Watermarking using Low Wavelet Subband based on 8×8 Sub-block DCT," in International Seminar on Application for Technology of Information and Communication (iSemantic), Semarang, 2017.
4. G. Ardiansyah, C. A. Sari, D. R. I. M. Setiadi and E. H. Rachmawanto, "Hybrid Method using 3-DES, DWT and LSB for Secure Image Steganography Algorithm," in International Conference on Information Technology, Information System, and Electrical Engineering (ICITISEE), Yogyakarta, 2017.
5. A. Setyono, D. R. I. M. Setiadi and Muljono, "StegoCrypt Method using Wavelet Transform and One-Time Pad for Secret Image Delivery," in International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE), Semarang, 2017.
6. A. U. Islam, F. Khalid, M. Shah, Z. Khan, T. Mahmood, A. Khan, U. Ali and M. Naeem, "An Improved Image Steganography Technique based on MSB using Bit Differencing," in International Conference on Innovative Computing Technology (INTECH), Dublin, 2016.