

SECRET MEDIA ENCRYPTION AND DECRYPTION OVER IMAGE

P SAI SANDEEP (17SS1A0438)

ALOKE VISHWAKARMA (17SS1A0403)

SEGGARI KOUSHIK (17SS1A0449)



Department of Electronics and Communication Engineering

Jawaharlal Nehru Technological University Hyderabad College of
Engineering Sultanpur.

Sultanpur(v),Pulka (M), Sangareddy-502273,Telangana

2020-2021

SECRET MEDIA ENCRYPTION AND DECRYPTION OVER IMAGE

A PROJECT REPORT

**SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
BACHELOR OF TECHNOLOGY IN
ELECTRONICS AND COMMUNICATION ENGINEERING
BY**

P SAI SANDEEP (17SS1A0438)

ALOKE VISHWAKARMA (17SS1A0403)

SEGGARI KOUSHIK (17SS1A0449)



Under the Guidance of

Dr. Y RAGHAVENDER RAO

Department of Electronics and Communication Engineering

**Jawaharlal Nehru Technological University Hyderabad
College of Engineering Sultanpur.
Sultanpur(V), pulkal(M), Sangareddy(Dist) – 502273, Telangana
2020-2021**

Jawaharlal Nehru Technological University Hyderabad
College of Engineering Sultanpur.
Sultanpur(V), pulkal (M), Sangareddy(Dist) – 502273, Telangana
2020-2021



Department of Electronics and Communication Engineering

CERTIFICATE

Date:

*This is to certify that the project report entitled **SECRET MEDIA ENCRYPTION AND DECRYPTION OVER IMAGE** is a bonafide work carried out by **P SAI SANDEEP, ALOKE VISHWAKARMA, SEGGARI KOUSHIK** bearing Roll No numbers **17SS1A0438, 17SS1A0403, 17SS1A0449** in partial fulfilment of the requirements for the degree of **BACHELOR OF TECHNOLOGY** in **ELECTRONICS & COMMUNICATION ENGINEERING** by the Jawaharlal Nehru Technological University, Hyderabad during the academic year 2020-2021.*

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

DR.Y.RAGHAVENDER RAO
Project Guide

DR.Y.RAGHAVENDER RAO
Head of the Department

ACKNOWLEDGMENT

We wish to take this opportunity to express our deep gratitude to all those who helped, encouraged, motivated and have extended their cooperation in various ways during our mini-project work. It is our pleasure to acknowledge the help of all those individuals who were responsible for foreseeing the successful completion of our project.

We express sincere gratitude to **Dr. B. BALU NAIK** Principal of JNTUHCES for his support during the course period.

We sincerely thank **Dr. V. VENKATESWARA REDDY** the Vice Principal of JNTUHCES for his kind help and cooperation.

We are thankful to **Dr. Y. RAGHAVENDER RAO**, Professor and Head of the Department of Electronics and Communication Engineering of JNTUHCES for his effective suggestions during the course period.

We would like to express our gratitude to **Sri V. RAJANESH**, Assistant Professor, for his guidance and constant supervision.

Finally, we would like to express our gratitude to **Mr T. MOHAN DAS**, for spending his valuable time and providing their guidance throughout the course period.

BY

P.SAI SANDEEP (17SS1A0438)

ALOK VISHWAKARMA (17SS1A0403)

SEGGARI KOUSHIK (17SS1A0449)

ABSTRACT

Technology is rapidly evolving in these modern times in conjunction with increased risk on data security. Data exchange over open channel is certainly not safe due to intruders and third-party attackers.

Steganography is the art and science of embedding hidden information into carrier media in such a way that no one apart from the sender and intended recipient even realizes that there is hidden message. This technique is intended to provides secrecy to the encrypted data. Many different carrier file formats can be used, but digital images are the most popular because of their frequency on the Internet. Different applications have different requirements of the steganography technique used. For example, some applications may require absolute invisibility of the secret information, while others require a larger secret message to be hidden.

This project intends to depict the **Encryption** of secret media (plain text, image, audio) into a carrier image to retain the privacy of the media and **decrypting** it back at the receiver's end. This is a simplest steganographic technique that embeds the bits of secret message directly into the least significant bit plane of the carrier image, so that the embedding procedure does not affect the original pixel value greatly. For a more secure approach, adding an extra layer of security to the project can be reliable and worthwhile. So, the message is encrypted using secret key and then sent to the receiver, so as to decrypts the message to get the original information at receiver's end.

KEYWORDS: Image processing, Encryption, Decryption, LSB Steganography, MATLAB

SOFTWARE: MATLAB

P. SAI SANDEEP (17SS1A0438)

ALOKE VISHWAKARMA (17SS1A0403)

SEGGARI KOUSHIK (17SS1A0449)

CONTENTS

CERTIFICATE	3
ACKNOWLEDGEMENT	4
ABSTRACT	5
LIST OF FIGURES	8
LIST OF TABLES	9
1. INTRODUCTION	10
1.1 INTRODUCTION	10
1.2 PROJECT DEFINITION	10
1.3 PROBLEM STATEMENT.....	11
1.4 PROJECT OBJECTIVES	11
1.5 ARCHITECTURE & COMPONENTS	11
1.6 PROJECT SCOPE	13
1.7 STEGANOGRAPHY ARCHITECTURE	13
1.8 STEGANOGRAPHY TYPES	14
2. LITERATURE REVIEW.....	16
2.1 INTRODUCTION	16
2.2 MOTIVATION	18
3. SYSTEM ANALYSIS AND DESIGN	19
3.1 STEGANOGRAPHY TECHNIQUES	19
3.2 FACTORS AFFECTING A STEGANOGRAPHIC METHOD	21
3.3 HOW IS IT DIFFERENT FROM CRYPTOGRAPHY	22
3.4 HOW STEGANOGRAPHY IS DONE	24
3.5 CONCEPT OF LSB BASED DATA EMBEDDING	25

3.6 EXTRACTION PROCESS	27
3.7 COMPARISON OF SECRET COMMUNICATION TECHNIQUES	28
4. SYSTEM DEVELOPMENT AND IMPLEMENTATION	29
4.1 MATLAB	29
4.1.1 INTRODUCTION	29
4.1.2 IMAGE PROCESSING	30
4.1.3 UNDERSTANDING THE IMAGE	31
4.2 MATLAB INSTALLATION	32
4.3 THE MATLAB SYSTEM	35
4.3.1 READING IMAGE IN MATLAB	36
4.3.2 DISPLAYING IMAGE	36
4.3.3 WRITING IMAGES	37
4.3.4 TYPECASTING IN MATLAB	38
4.4 MATLAB GUI	38
5. RESULT AND CONCLUSION	44
5.1 ENCRYPTION OF DATA.....	44
5.2 DECRYPTION OF DATA.....	47
5.3 APPLICATIONS OF IMAGE STEGANOGRAPHY	47
5.4 LIMITATIONS	49
5.5 CONCLUSION	50
5.6 FUTURE SCOPE	50
REFERENCES	51
APPENDIX	52

LIST OF FIGURES

FIGURE	PAGE NO
FIG 1.1 PROCESS OF STEGANOGRAPHY.....	12
FIG 1.2 STEGANOGRAPHY ALGORITHM.....	13
FIG 3.1 STEGANOGRAPHIC SYSTEM.....	19
FIG 3.2 PROCESS OF IMAGE STEGANOGRAPHY.....	24
FIG 3.3 COVER IMAGE.....	26
FIG 3.4 ENCRYPTED IMAGE.....	26
FIG 3.5 ORIGINAL IMAGE VS STEGO IMAGE.....	27
FIG 3.6 SCREENSHOT OF PIXEL VALUES OF IMAGE.....	28
FIG 4.1 MATLAB LOGO.....	29
FIG 4.2 MATLAB INSTALLATION PROMPT IN WINDOWS.....	32
FIG 4.3 CHOOSE INSTALLATION FOLDER PROMPT.....	33
FIG 4.4 MATLAB HOMEPAGE GUI.....	34
FIG 4.5 IMAGE DISPLAYED USING IMSHOW().....	36
FIG 4.6 MATLAB GUIDE PROMPT.....	39
FIG 4.7 EMPTY GUI	40
FIG 4.8 GUI CREATED WITH MATLAB	40
FIG 4.9 FULLY FUNCTIONAL GUI	41
FIG 4.10 ORIGINAL IMAGE	42
FIG 4.11 TEXT AND IMAGE	42
FIG 4.12 FINAL GUI	43

FIG 5.1 MATLAB GUI	44
FIG 5.2 TEXT FILE	45
FIG 5.3 COVER IMAGE	45
FIG 5.4 LSB STEGANOGRAPHY ALGORITHM	46
FIG 5.5 ENCRYPTED IMAGE	46
FIG 5.6 DECRYPTED TEXT MESSAGE	47

LIST OF TABLES

TABLE 3.1 STEGANOGRAPHY VS CRYPTOGRAPHY	23
TABLE 3.2 COMPARISION OF VARIOUS SECRET COMMUNICATION TECHINIQUES	28

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The growing use of Internet needs to take attention while we send and receive personal information in a secured manner. For this, there are many approaches that can transfer the data into different forms so that their resultant data can be understood if it can be returned back into its original form. This technique is known as encryption. However, a major disadvantage of this method is that the existence of data is not hidden. If someone gives enough time then the unreadable encrypted data may be converted into its original form.

In this document, we propose a new system for hiding data stands on many methods and algorithms for image hiding where we store on data file, called sink file in an image file called as container image. The primary objective is to use steganography techniques so as to provide more security and simultaneously using less storage.

1.2 PROJECT DEFENITION

In this project, we propose to develop a system to hiding data by using "STEGANOGRAPHY" technique as we used many methods stands on some techniques to have at the back-end a software for hiding data based on hiding algorithms. After studying the data hiding algorithms we found many ways to hiding data by using the multimedia files and the main question for me was "Where hidden data hides?" as we found by our search to know where the data hides it's important to know what is the file type of the data that it shall be hidden and the cover file type so it is possible to alter graphic or sound files slightly without losing their overall viability for the viewer and listener. With audio, you can use bits of file that contain sound not audible to the human ear. With graphic images, you can remove redundant bits of color from the image and still produce a picture that looks intact to human eye and is difficult to discern from its original. It is in those bits that stego hides its data.

By the final of our project we developed a GUI which uses an algorithm, to embed data in an image; The purposed system is called "Steganography", the aim of this project is to encrypt the data; the meaning of encrypt is to hide the data over an image using different steganographic algorithms, in this system LSB is the algorithms that we use to hiding the data.

1.3 PROBLEM STATEMENT

This project addresses the security problem of transmitting the data over internet network, the main idea coming when we start asking that how can we send a message secretly to the destination? The science of steganography answers this question. Using steganography, information can be hidden in carriers such as images, audio files, text files, videos and data transmissions. In this document, we proposed some methods and algorithms of an image steganography system to hide a digital text of a secret message.

1.4 PROJECT OBJECTIVES

In this project we primarily concentrated on the data security issues when sending the data over the network using steganographic techniques. The main objectives of our project are to product security tool based on steganography techniques to hider message carried by stego-media which should not be sensible to human beings and avoid drawing suspicion to the existence of hidden message.

1.5 ARCHITECTURE & COMPONENTS

In the proposed system we concentrate on finding some algorithm to hide the data inside images using steganography technique. An algorithm is designed to hide all the data inputted within the image to protect the privacy of the data. Then, the system is developed based on the new steganography algorithm. This proposed system provides the user with two options encrypt and decrypt the data, in encryption the secret information is hiding in with image file, and on the other side the decryption is getting the hidden information from the stego image file, and also the user can show the image size after and before the encryption.

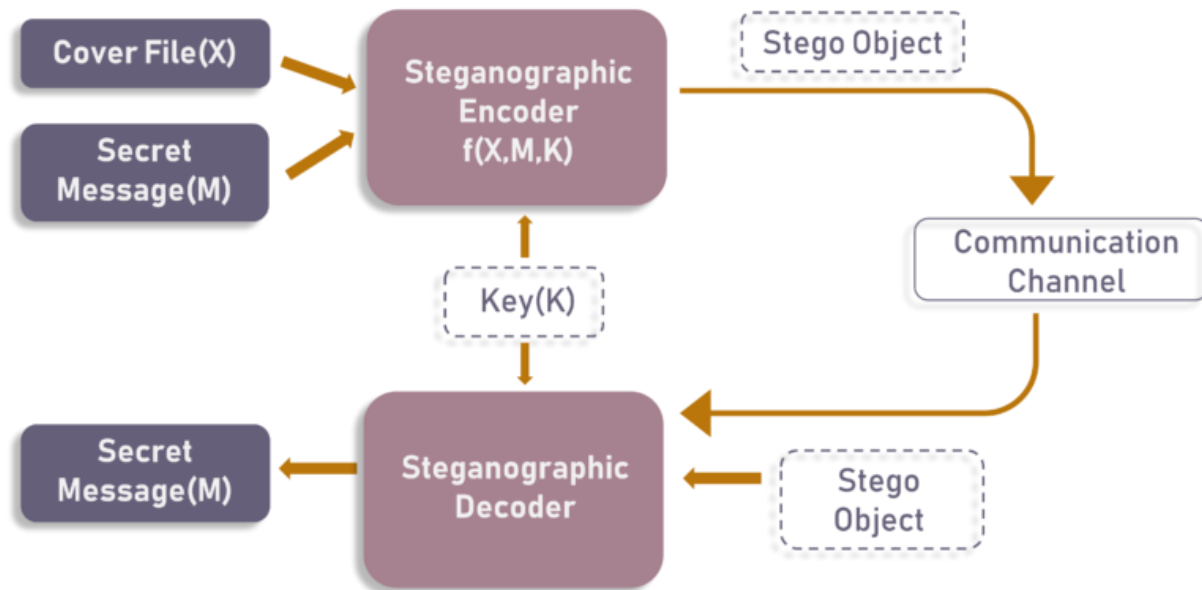


Fig 1.1 – Process of steganography

The processes of encryption and decryption of the data file consists of:

- Providing security for the data to be transmitted through network using steganography.
- Proposing an approach for hiding the data within an image using a steganographic algorithm which provides better accuracy and quality of hiding.

Matlab's image processing capability is used to extensively analyze the functions of the LSB algorithm in steganography. Texts and other file formats are encrypted and embedded into an image file which is then transferred to the destination.

1.6 PROJECT SCOPE

Our project scope is developed for hiding information in any image file to ensure the safety of exchange the data between different parties and provide better security during message transmission. The scope of the project is implementation of steganography tools for hiding information includes any type of information file and image files and the path where the user wants to save image and extruded file. We will use LSB technique; the proposed approach is to

use the suitable algorithm for embedding the data in an image files; we will show a brief of this algorithm that we used to hiding data.

1.7 STEGANOGRAPHY ARCHITECTURE

Steganography is the art of hiding and transmitting data through apparently innocuous carriers in an effort to conceal the existence of the data, the word Steganography literally means covered or hiding writing as derived from Greek. Steganography has its place in security. It is not intended to replace cryptography but supplement it. [1] Hiding a message with Steganography methods reduces the chance of a message being detected. If the message is also encrypted then it provides another layer of protection.

Therefore, some Steganographic methods combine traditional Cryptography with Steganography; the sender encrypts the secret message prior to the overall communication process, as it is more difficult for an attacker to detect embedded cipher text in a cover. It has been used through the ages by ordinary people, spies, rulers, government, and armies. There are many stories about Steganography. [1] For example, ancient Greece used methods for hiding messages such as hiding in the field of Steganography, some terminology has developed. The adjectives 'cover', 'embedded', and 'stego' were defined at the information hiding workshop held in Cambridge, England. The term "cover" refers to description of the original, innocent message, data, audio, video, and so on. Steganography is not a new science; it dates back to ancient times.

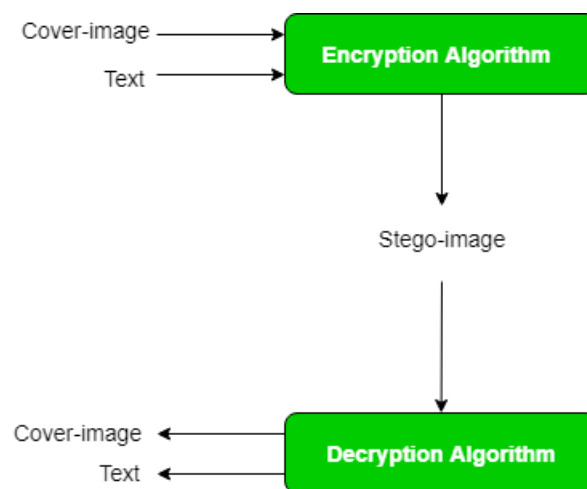


Fig 1.2 - Steganography algorithm

Hidden information in the cover data is known as the "embedded" data and information hiding is a general term encompassing many sub disciplines, is a term around a wide range of problems beyond that of embedding message in content. The term hiding here can refer to either making the information undetectable or keeping the existence of the information secret. [1] Information hiding is a technique of hiding secret using redundant cover data such as images, audios, movies, documents, etc. This technique has recently become important in a number of application areas. For example, digital video, audio, and images are increasingly embedded with imperceptible marks, which may contain hidden signatures or watermarks that help to prevent unauthorized copy. It is a performance that inserts secret messages into a cover file, so that the existence of the messages is not apparent. [1]

Research in information hiding has tremendous increased during the past decade with commercial interests driving the field. Although the art of concealment "hidden information" as old as the history, but the emergence of computer and the evolution of sciences and techniques breathe life again in this art with the use of new ideas, techniques, drawing on the computer characteristics in the way representation of the data, well-known computer representation of all data including (Multimedia) is binary these representations are often the digital levels and areas and change values-aware of slight not aware or felt by Means sensual of human such as hearing, sight, the advantage use of these properties to hide data in multimedia by replace the values of these sites to the values of data to be hidden, taking into account the acceptable limits for the changeover, and not exceeded to prevent degradation media container with a change becomes aware and felt by human. It should be noted here that although the art of hidden information come in the beginning of the computer and its techniques However, the seriousness of the work in the stenography as a stand-alone science started in 1995. [1]

1.8 STEGANOGRAPHY TYPES

Steganography is derived from the two Greek words Stego and Graphia, Stego means covering and graphia which means writing, thus the translation is covered writing or the hiding the data. The simplest way to do this process is by inserting the confidential data bits in LSB positions of original image.

Types of Steganography:

- Image to image
- Text to image
- Image to text
- Video to voice
- Voice to video

Image to Image: - In the image steganography the image is inserted within another image by using the stego key.

Text to Image: - In this, the text is inserted within the image and sends the image with the help of the symmetric key.

Video to voice: - Hiding or embedding message in the video is like an art of hiding information because the sender is not only hiding but how that message is prevented open by anyone except receiver. Hiding message in the video is part of the art of hiding information, Video-based steganography techniques are same like image based In Today 's world, Information security is the sensitive case of security and it getting necessary to protect the data from being tampered. For the data not being tampered, we are using this steganography technique. The protection should be required, when the data is already placed at the transmission.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

In this chapter, we will provide an overview of steganography using LSB to hide the files inside images using MATLAB. It is development environment is used in a variety of domains, such as image and signal processing, MATLAB offers many “toolboxes”, and a simple interface to high-performance libraries, one of the most advantages is a large user community with lots of free code and knowledge sharing and the ability to process both still images and video. It is popular because of its ease and simplicity. Also, we will mention some programs that have the same approach it is using an encryption. Then, we will give the recommendation that help to develop our program.

In this project, we use a method of encrypting any data file in an image file. This process of hiding the data helps to sharing the information with others over the internet network without any potential risk. The proposed system will help to hide the content with in the image and encryption of data file with in the image will help to make the document much securer. In this project, we developed the proposed system by using steganographic algorithm which is LSB and a technique for hiding capacity and efficiency of hiding the message with in an image.

In 2010, Jing-Ming Guo, Member, IEEE, and Thanh-Nam Le proposed a paper on “Secret Communication Using JPEG Double Compression” which says protecting privacy for exchanging information through the media has been a topic researched by many people. Up to now, cryptography has always had its ultimate role in protecting the secrecy between the sender and the intended receiver. However, nowadays steganography techniques are used increasingly besides cryptography to add more protective layer to the hidden data. In this paper they showed that the quality factor in a JPEG image can be an embedding space, and they discussed the ability of embedding a message to a JPEG image by managing JPEG quantization tables (QTs). In combination with some permutation algorithms, this scheme can be used as a tool for secret

communication. The proposed method can achieve satisfactory decoded results with this straightforward JPEG double compression strategy.

In 2011, Wei-Jen Wang, Cheng-Ta Huang, and Shih-Jeng Wang proposed a paper on “Data hiding is one of the most important techniques to achieve better data and communication protection by hiding information into a media carrier. It provides a secure method to distribute data through a public and open channel. Data hiding for vector quantization (VQ)-based images focuses on the problem of embedding secret data into a cover VQ-based image to achieve secret communication and data protection. This paper provides a state-of-the-art review and comparison of the different existing data-hiding methods for VQ-based images. In this paper, we classify VQ-based data-hiding methods into four no overlapping groups according to their reversibility and output formats, introduce the details of the representative methods, summarize the features of the representative methods, and compare the performance of the representative methods using peak signal-to-noise ratio, capacity of secret data, and bit rate. This paper shows that an irreversible method is very likely a VQ-based data-hiding method that produces a stego-image as its output, and it can embed more secret data than a reversible method. Nonstandard encoding methods (e.g., joint neighbouring coding) are becoming popular in reversible data hiding since they can increase the capacity for embedding the secret data.

In 2012, Fangjun Huang, Member, IEEE, Jiwu Huang, Senior Member, IEEE, and Yun-Qing Shi, Fellow, IEEE” Suggest a paper, which presents a new channel selection rule for joint photographic experts’ group (JPEG) steganography, which can be utilized to find the discrete cosine transform (DCT) coefficients that may introduce minimal detectable distortion for data hiding. Three factors are considered in our proposed channel selection rule, i.e., the perturbation error (PE), the quantization step (QS), and the magnitude of quantized DCT coefficient to be modified (MQ). Experimental results demonstrate that higher security performance can be obtained in JPEG steganography via our new channel selection rule.”

In the year of 2013 Akhtar, N.; Johri, P.; Khan, S., implemented a variation of plain LSB (Least Significant Bit) algorithm. The stego-image quality has been improved by using bit-inversion technique. LSB method improving the PSNR of stegoimage. Through storing the bit patterns for which LSBs are inverted, image may be obtained correctly. For the improving the robustness of steganography, RC4 algorithm had been implemented to achieve the randomization in hiding

message image bits into cover image pixels instead of storing them sequentially. This method randomly disperses the bits of the message in the cover image and thus, harder for unauthorized people to extract the original message. The presented method shows good enhancement to Least Significant Bit technique in consideration to security as well as image quality.

In the year of 2013 Prabakaran, G.; Bhavani, R. and Rajeswari P.S. Investigated on Medical records are extremely sensitive patient information a multi secure and robustness of medical image based steganography scheme is proposed. This methodology provides an efficient and storage security mechanism for the protection of digital medical images. Authors proposed a viable steganography method using Integer Wavelet Transform to protect the MRI medical image into a single container image. The patient's medical diagnosis image has been taken as secret image and Arnold transform was applied and scrambled secret image was obtained. In this case, the scrambled secret image was embedded into the dummy container image and Inverse IWT was taken to get a dummy secret image. It has been observed that the quality parameters are improved with acceptable PSNR compared to the existing algorithm.

2.2 MOTIVATION

In steganography, the message is embedded into the digital media rather than encrypting it. The digital media contents, called the cover, can be determined by anybody, the message hidden in the cover can be detected by the one having the true key. The message in the message after the receiver gets the data. That allows steganography to protect the embedded information after it is decrypted. Steganography is therefore broader than cryptography. Signal processing area includes- filtering, de-noising method, interference suppression, radar signal processing, electromagnetic wave propagation, and wireless communication systems. The area of the image processing applications includes steganography, watermarking .

CHAPTER 3

SYSTEM ANALYSIS AND DESIGN

Steganography consists of two terms that is message and cover image. Message is the secret data that needs to hide and cover image is the carrier that hides the message in it. Secret Data
Cover Image Data Embedding Stego- Image Algorithm

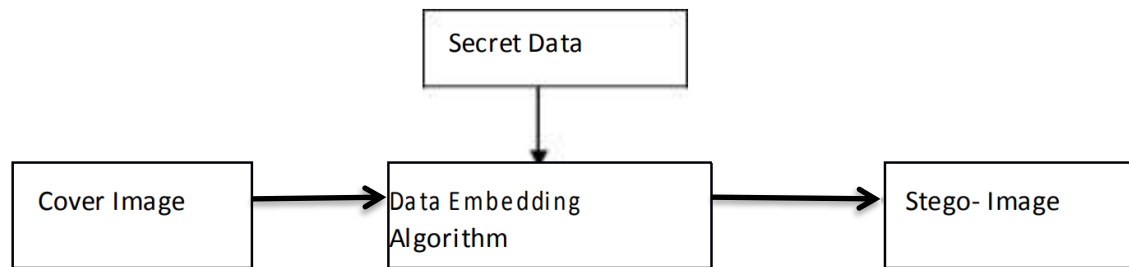


Fig 3.1 – Steganographic system

3.1 STEGANOGRAPHY TECHNIQUES:

1. Spatial Domain Methods: In this method the secret data is embedded directly in the intensity of pixels. It means some pixel values of the image are changed directly during hiding data. Spatial domain techniques are classified into following categories:

i)Least significant bit (LSB) ii) Pixel value differencing (PVD) iii) Edges based data embedding method (EBE) iv) Random pixel embedding method (RPE) v)Mapping pixel to hidden data method vi) Labelling or connectivity method vii) Pixel intensity based.

i) LSB: this method is most commonly used for hiding data. In this method the embedding is done by replacing the least significant bits of image pixels with the bits of secret data. The image obtained after embedding is almost similar to original image because the change in the LSB of image pixel does not bring too much differences in the image.

ii) BPCP: In this segmentation of image are used by measuring its complexity. Complexity is used to determine the noisy block. In this method noisy blocks of bit plan are replaced by the binary patterns mapped from a secret data

iii) PVD: In this method, two consecutive pixels are selected for embedding the data. Payload is determined by checking the difference between two consecutive pixels and it serves as basis for identifying whether the two pixels belongs to an edge area or smooth area.

2. Spread Spectrum Technique : The concept of spread spectrum is used in this technique. In this method the secret data is spread over a wide frequency bandwidth. The ratio of signal to noise in every frequency band must be so small that it become difficult to detect the presence of data. Even if parts of data are removed from several bands, there would be still enough information is present in other bands to recover the data. Thus it is difficult to remove the data completely without entirely destroying the cover .It is a very robust technique mostly used in military communication.

3. Statistical Technique : In the technique message is embedded by changing several properties of the cover. It involves the splitting of cover into blocks and then embedding one message bit in each block. The cover block is modified only when the size of message bit is one otherwise no modification is required.

4. Transform Domain Technique: In this technique; the secret message is embedded in the transform or frequency domain of the cover. This is a more complex way of hiding message in an image. Different algorithms and transformations are used on the image to hide message in it. Transform domain techniques are broadly classified such as Discrete Fourier transformation technique (DFT) ii) Discrete cosine transformation technique (DCT) iii) Discrete Wavelet transformation technique (DWT) iv) Lossless or reversible method (DCT) iv)Embedding in coefficient bits

5. Distortion Techniques: In this technique the secret message is stored by distorting the signal. A sequence of modification is applied to the cover by the encoder. The decoder measures the differences between the original cover and the distorted cover to detect the sequence of modifications and consequently recover the secret message.

6. Masking and Filtering : These techniques hide information by marking an image. Steganography only hides the information where as watermarks becomes a portion of the image. These techniques embed the information in the more significant areas rather than hiding it into the noise level. Watermarking techniques can be applied without the fear of image destruction due to loss y compression as they are more integrated into the image. This method is basically used for 24-bit and grey scale images.

3.2 FACTORS AFFECTING A STENOGRAPHIC METHOD

The effectiveness of any stenographic method can be determined by comparing stego –image with the cover Image.

There are some factors that determines the efficiency of a technique. These factors are:

1) Robustness: Robustness refers to the ability of embedded data to remain intact if the stego – image undergoes transformations, such as linear and non-linear filtering, sharpening or blurring, addition of random noise, rotations and scaling, cropping or decimation, lossy compression.

2) Imperceptibility: The imperceptibility means invisibility of a steganographic algorithm. Because it is the first and foremost requirement, since the strength of steganography lies in its ability to be unnoticed by the human eye.

3) PSNR (Peak Signal to Noise Ratio): It is defined as the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. This ratio measures the quality between the original and a compressed image. The higher value of PSNR represents the better quality of the compressed image.

4) MSE (Mean Square Error): It is defined as the average squared difference between a reference image and a distorted image. The smaller the MSE, the more efficient the image steganography technique . MSE is computed pixel-by-pixel by adding up the squared differences of all the pixels and dividing by the total pixel count.

5) SNR (Signal to Noise Ratio): It is the ratio between the signal power and the noise power. It compares the level of a desired signal to the level of background noise.

3.3 HOW IS IT DIFFERENT FROM CRYPTOGRAPHY?

Cryptography and steganography are both methods used to hide or protect secret data. However, they differ in the respect that cryptography makes the data unreadable, or hides the *meaning* of the data, while steganography hides the *existence* of the data.

In layman's terms, cryptography is similar to writing a letter in a secret language: people can read it, but won't understand what it means. However, the existence of a (probably secret) message would be obvious to anyone who sees the letter, and if someone either knows or figures out your secret language, then your message can easily be read.

If you were to use *steganography* in the same situation, you would hide the letter inside a pair of socks that you would be gifting the intended recipient of the letter. To those who don't know about the message, it would look like there was nothing more to your gift than the socks. But the intended recipient knows what to look for, and finds the message hidden in them.

Similarly, if two users exchanged media files over the internet, it would be more difficult to determine whether these files contain hidden messages, than if they were communicating using cryptography.

Cryptography is the process used for the conversion of the plain text into cipher text by using the symmetric key and this process is known as the encryption. The main disadvantage of the cryptography is that the plaintext can be known and the cipher text is visible but we can't read it. Steganography is a method that the plain text is concealed into the digital media. In this process the Trespasser can't be able to see the plaintext or the cipher text because it is concealing into another media. The trespasser can't suspect if there is any confidential data that is existing. The steganography technique is used for the better security of the data over the computer network. Cryptography is often used to supplement the security offered by steganography. Cryptography algorithms are used to encrypt secret data before embedding it into cover files.

Table 3.1 – Steganography vs cryptography

S.NO	Steganography	Cryptography
1.	Steganography means covered writing .	Cryptography means secret writing .
2.	Steganography is less popular than Cryptography.	While cryptography is more popular than Steganography.
3.	Attack's name in Steganography is Steganalysis .	While in cryptography, Attack's name is Cryptanalysis .
4.	In steganography, structure of data is not usually altered.	While in cryptography, structure of data is altered.
5.	Steganography supports Confidentiality and Authentication security principles.	While cryptography supports Confidentiality and Authentication security principles as well as Data integrity and Non-repudiation .
6.	In steganography, the fact that a secret communication is taking place is hidden.	While in cryptography only secret message is hidden.
7.	In steganography, not much mathematical transformations are involved.	Cryptography involves the use of number theory, mathematics etc. to modify data

3.4 HOW STEGANOGRAPHY IS DONE?

An image is represented as an $N \times M$ (in case of greyscale images) or $N \times M \times 3$ (in case of colour images) matrix in memory, with each entry representing the intensity value of a pixel. In image steganography, a message is embedded into an image by altering the values of some pixels, which are chosen by an encryption algorithm. The recipient of the image must be aware of the same algorithm in order to know which pixels he or she must select to extract the message.

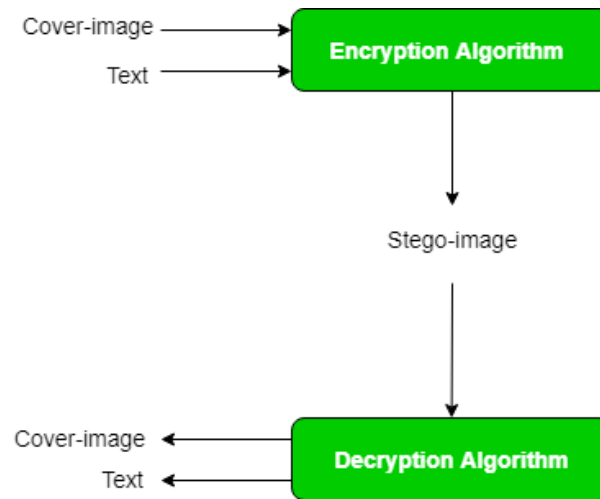


Fig 3.2 – Process of Image Steganography

Detection of the message within the cover-image is done by the process of **steganalysis**. This can be done through comparison with the cover image, histogram plotting, or by noise detection. While efforts are being invested in developing new algorithms with a greater degree of immunity against such attacks, efforts are also being devoted towards improving existing algorithms for steganalysis, to detect exchange of secret information between terrorists or criminal elements.

The main idea behind steganography is to hide the existence of data in any medium like audio, video, image, etc. When we talk about image steganography, the idea is quite simple. Images are made up of pixels which usually refer to the color of that particular pixel. In a greyscale

(black and white) image, these pixel values range from **0-255**, 0 being black and 255 being white.

3.5 CONCEPT OF LSB BASED DATA EMBEDDING:

LSB stands for Least Significant Bit. The idea behind LSB embedding is that if we change the last bit value of a pixel, there won't be much visible change in the color. For example, 0 is black. Changing the value to 1 won't make much of a difference since it is still black, just a lighter shade.

The encoding is done using the following steps:

1. Convert the image to greyscale
2. Resize the image if needed
3. Convert the message to its binary format
4. Initialize output image same as input image
5. Traverse through each pixel of the image and do the following:
 - Convert the pixel value to binary
 - Get the next bit of the message to be embedded
 - Create a variable **temp**
 - If the message bit and the LSB of the pixel are same, set $\text{temp} = 0$
 - If the message bit and the LSB of the pixel are different, set $\text{temp} = 1$
 - This setting of temp can be done by taking XOR of message bit and the LSB of the pixel
 - Update the pixel of output image to input image pixel value + **temp**
6. Keep updating the output image till all the bits in the message are embedded
7. Finally, write the input as well as the output image to local system.

EXAMPLE:

Input : message='AlbertEinstein' 



Fig 3.3 - Cover image

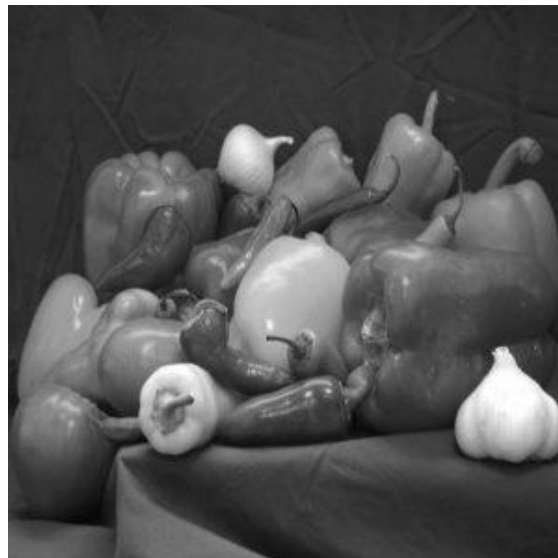
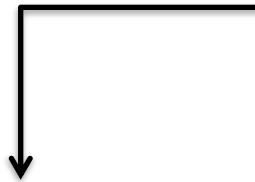


Fig 3.4 - Encrypted image

3.6 EXTRACTION PROCESS:

The extraction process is simple. We need to first calculate how many pixels is the text stored in. For example, the text “**AlbertEinstein**” has 13 characters. Each character is represented in 8 bits. So, the number of pixels in which the text is stored will be $13 * 8 = 104$. Now after knowing this, we need to traverse through the image, one pixel at a time. We store the Least Significant Bit (LSB) of each pixel in an array *extracted bits*.

After extracting the LSBs of the required pixels, we need to take every 8 bits from *extracted bits* and convert it to the corresponding character. In this way, the text stored in the **stego image** can be extracted.

In this method, we take the pixel values of the image obtained in the previous method. The values are stored in **xlsx** format. The message embedded in the image is “**AlbertEinstein**”. Here is a screenshot of the input image and the stego image obtained from the prerequisite article:

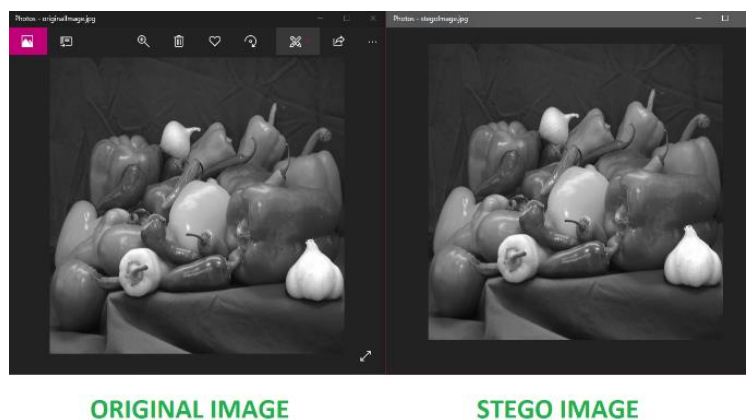


Fig 3.5 – original vs stego image

Input : A screenshot of the pixel values of the image:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	44	45	47	44	42	45	45	43	44	45	45	42	44	45	44	43	44	47	45	44
2	44	44	45	44	42	43	44	44	44	44	44	43	43	44	44	44	45	45	44	44
3	44	44	44	44	43	43	44	45	45	44	44	44	44	44	44	45	45	45	44	44
4	43	44	44	44	45	44	43	45	45	45	45	45	46	45	45	46	46	45	44	45
5	43	44	45	44	44	44	45	46	46	47	46	45	45	47	47	48	46	46	45	46
6	44	45	46	44	45	44	46	45	46	46	45	46	46	47	47	48	45	45	46	46
7	45	44	45	44	45	43	46	45	45	45	44	46	47	46	47	46	45	45	46	46
8	45	43	44	45	44	42	45	44	44	46	44	45	47	45	46	45	45	46	46	47
9	43	43	44	45	44	43	44	44	46	46	46	47	46	46	46	45	47	47	47	47
10	44	44	45	44	44	45	45	43	45	44	45	46	45	46	46	45	47	48	46	46
11	44	44	44	44	45	46	45	43	45	43	44	44	44	46	46	45	46	47	46	46
12	44	44	43	45	46	46	45	43	45	43	45	43	44	45	45	45	45	44	45	45
13	42	44	44	45	45	44	44	44	43	45	46	47	44	45	45	45	45	42	44	46
14	41	43	44	46	46	46	44	43	44	46	47	47	45	45	45	44	44	42	44	46
15	41	41	43	46	46	47	44	43	44	46	47	46	46	45	45	43	44	44	45	46
16	42	41	42	45	46	47	44	43	44	45	46	46	47	46	44	43	45	46	46	46
17	43	43	41	43	44	44	45	44	43	44	45	45	46	45	43	45	46	45	47	46
18	42	44	44	43	42	44	45	45	44	44	45	45	44	44	44	46	45	43	45	45
19	43	44	45	45	41	44	45	47	45	44	45	45	44	44	45	46	44	43	44	45
20	44	44	45	45	42	43	44	47	45	43	44	45	44	44	45	45	44	44	44	44
21	44	44	43	43	44	46	43	44	43	44	45	45	43	44	43	42	42	43	43	45
22	42	43	42	43	44	45	42	42	43	46	46	45	45	45	43	42	42	43	43	45

Fig 3.6 – screenshot of pixel values of the image

Output: AlbertEinstein

3.7 COMPARISON OF SECRET COMMUNICATION TECHNIQUES.

Table 3.2 – Comparison of various secret communication techniques

Secret Communication Techniques	Confidentiality	Integrity	Un removability
Encryption	Yes	No	Yes
Digital Signatures	No	Yes	No
Steganography	Yes/No	Yes/No	Yes

CHAPTER 4

SYSTEM DEVELOPMENT AND IMPLEMENTATION

TOOLS USED

1. MATLAB

4.1 MATLAB

4.1.1 INTRODUCTION

MATLAB (an abbreviation of "matrix laboratory") is a proprietary multi-paradigm programming language and numeric computing environment developed by MathWorks. MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages.

It is a programming platform designed specifically for engineers and scientists. The heart of MATLAB is the MATLAB language, a matrix-based language allowing the most natural expression of computational mathematics.

As of 2020, MATLAB has more than 4 million users worldwide. MATLAB users come from various backgrounds of engineering, science, and economics. 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.

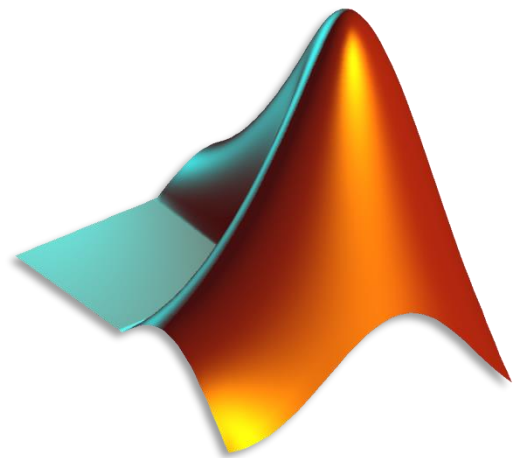


Fig 4.1 - MATLAB logo

Typical uses include:

- Math and computation
- Algorithm development
- Modelling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including GUI build

The language, apps, and built-in math functions enable you to quickly explore multiple approaches to arrive at a solution. MATLAB lets you take your ideas from research to production by deploying to enterprise applications and embedded devices, as well as integrating with Simulink® and Model-Based Design.

Millions of engineers and scientists in industry and academia use MATLAB. You can use MATLAB for a range of applications, including deep learning and machine learning, signal processing and communications, image and video processing, control systems, test and measurement, computational finance, and computational biology.

MATLAB features a family of application-specific solutions called toolboxes. Very important to most users of MATLAB, toolboxes allow you to *learn* and *apply* specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems and others.

4.1.2 IMAGE PROCESSING: -

In computer science, **digital image processing** is the use of computer algorithms to perform image processing on digital images. As a subcategory or field of digital signal processing, digital image processing has many advantages over analog image processing. It allows a much wider range of algorithms to be applied to the input data and can avoid problems such as the build-up

of noise and signal distortion during processing. Since images are defined over two dimensions (perhaps more) digital image processing may be modeled in the form of multidimensional systems.

Digital image processing allows the use of much more complex algorithms, and hence, can offer both more sophisticated performance at simple tasks, and the implementation of methods which would be impossible by analog means.

In particular, digital image processing is the only practical technology for:

1. Classification
2. Feature extraction
3. Multi-scale signal analysis
4. Pattern recognition
5. Projection

4.1.3 UNDERSTANDING THE IMAGE

It is defined as a two dimensional function $f(x,y)$, where x and y are spatial coordinates. Amplitude of f at any pair of coordinates (x,y) is called intensity or gray level of image at that point. When (x,y) and amplitudes values of f are all finite, discrete quantities, image is said to be digital image.

Hence digital image processing refers to processing digital images by means of a digital computer.

A digital image is composed of a finite number of elements are referred to as picture elements, pels and pixels. Pixel is the most widely used term to denote the elements of an image.

Moreover, digital image processing encompasses processes whose inputs and outputs are images and in addition, encompasses processes that extract attributes from images up to and including the recognition of individual objects.

4.2 MATLAB Installation

Step 1: Double click on the MATLAB icon (the binary file which we downloaded earlier). After clicking the icon, a pop-up will ask for the installer to run, click on the *Run*. A MathWorks Installer window will pop-up on the screen.

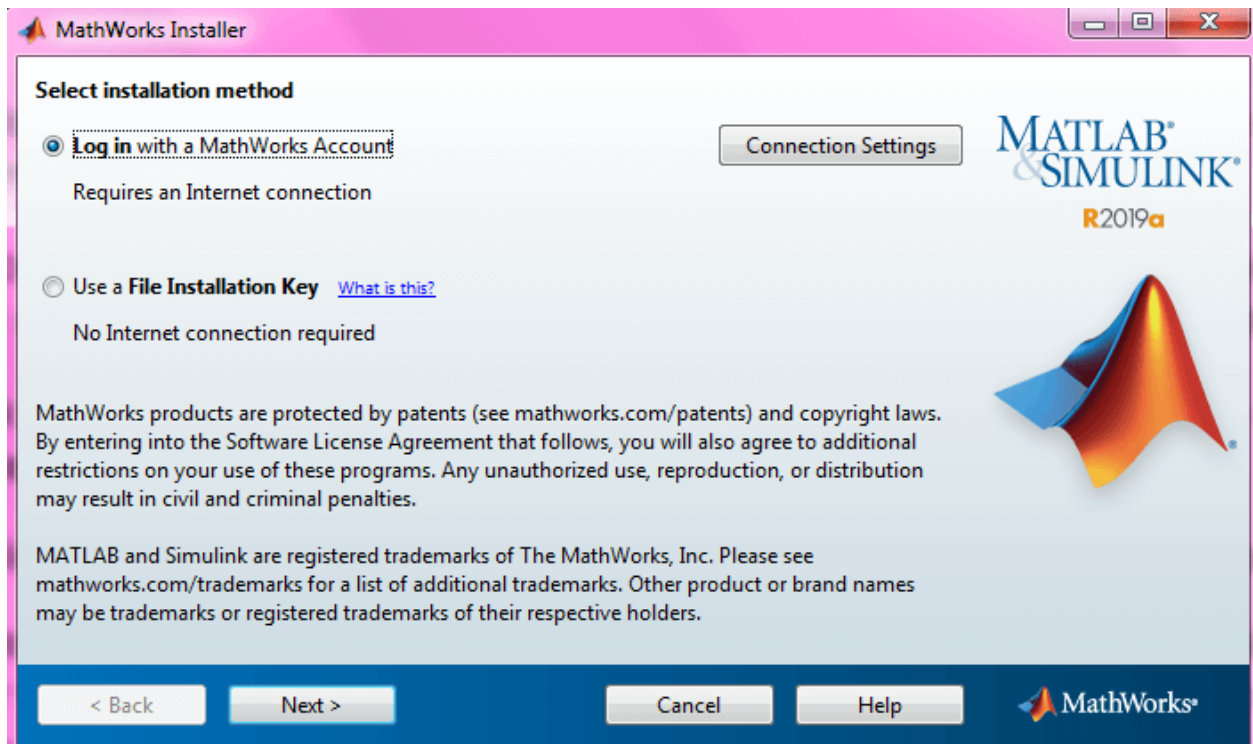


Fig 4.2 – MATLAB installation prompt in windows

- Click on **Log in with a MathWorks Account** and click next.

Step 2: A license Selection window will appear, a preselected license id will be highlighted with a blue background. Here you have to select your license id; this is the id which we have saved during STEP 9 of downloading of the installer (we urged to note down that id during that time) and again click on Next.

- A new Folder Selection window appears, no need to change the folder location for installation of MATLAB, click on Next.

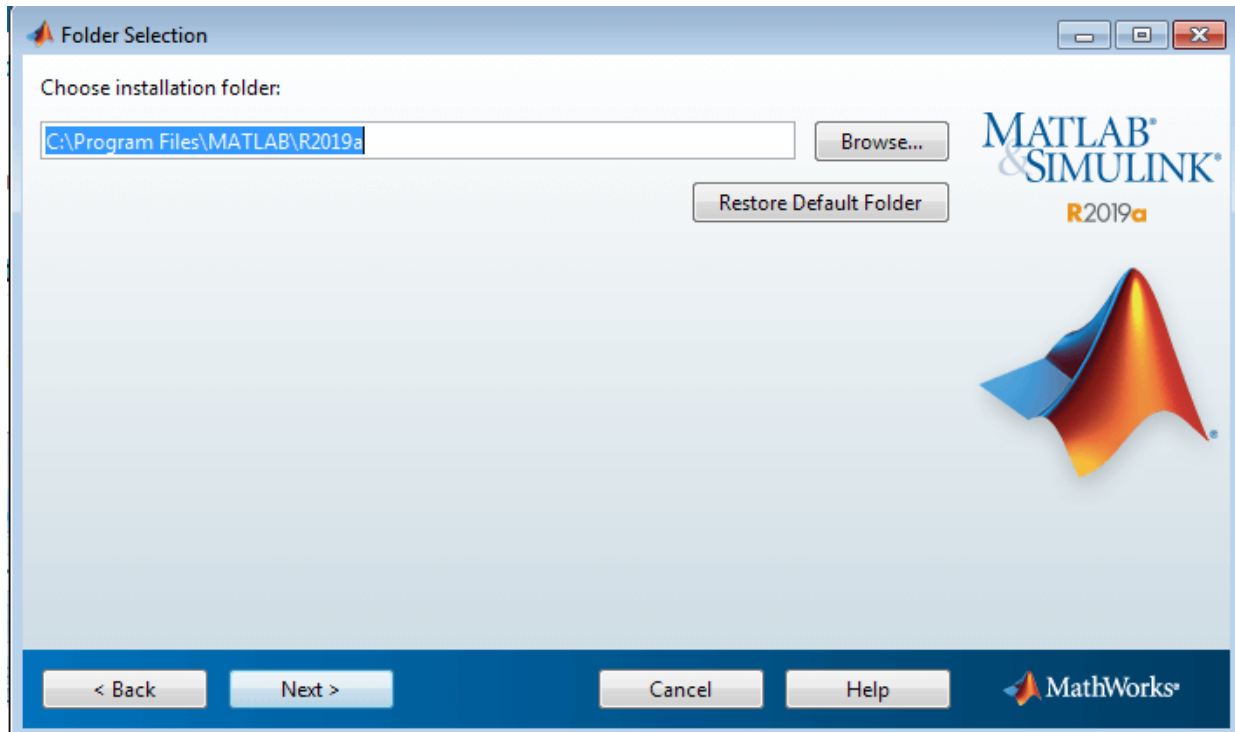


Fig 4.3 – Choose installation folder prompt

Step 3: Next is Product Selection window, the first product is MATLAB 9.6, this is mandatory to select because it is the MATLAB environment, and from other products, you can choose as many of your choices and click on Next.


- Next is the *Installation Options* window, select options as per your choice. Any time you feel something to change, you can go back to the previous step by clicking on the Back button.
- Next is the Confirmation window, here you no need to do anything, confirm what you are going to download in the process of the installation of MATLAB, its other Add-on products, and what is the size of the downloads; and click on Install.
- By clicking on Install, downloading of all the products will be started. It's a massive download, so you have to wait for some time to complete the download.

Step 4: After downloading of all products and completion of the installation, a window appears that says to Activate the MATLAB, no need to do anything, click on the Next button.

- After clicking on Next, a new window appears that says about what is the meaning of

activation. Proceed by clicking on *Next*.

- Again a new window appears displaying your email id and your products' license id, proceed by clicking on *Confirm button*.
- Congratulations, you have completed the installation process and successfully installed the MATLAB and its other products. Now click on the *Finish button*.

Step 5: A MATLAB shortcut will be created on the desktop as per our choice during the installation process. Now we can work with MATLAB by clicking on the icon  placed there on the desktop.

MATLAB provides a Desktop GUI based environment. The Home tab of the environment contains three panels there:

1. Current folder: It is located on the left side; here, you can access your files.
2. Command Window: It is located on the right side; it is the command prompt, and here you can enter commands to operate the functions, to assign variables, and for calculations.
3. Workspace: It is located on the left side right below the Current Folder; here, all variables that you create are stored, and data from other files can also be imported here.

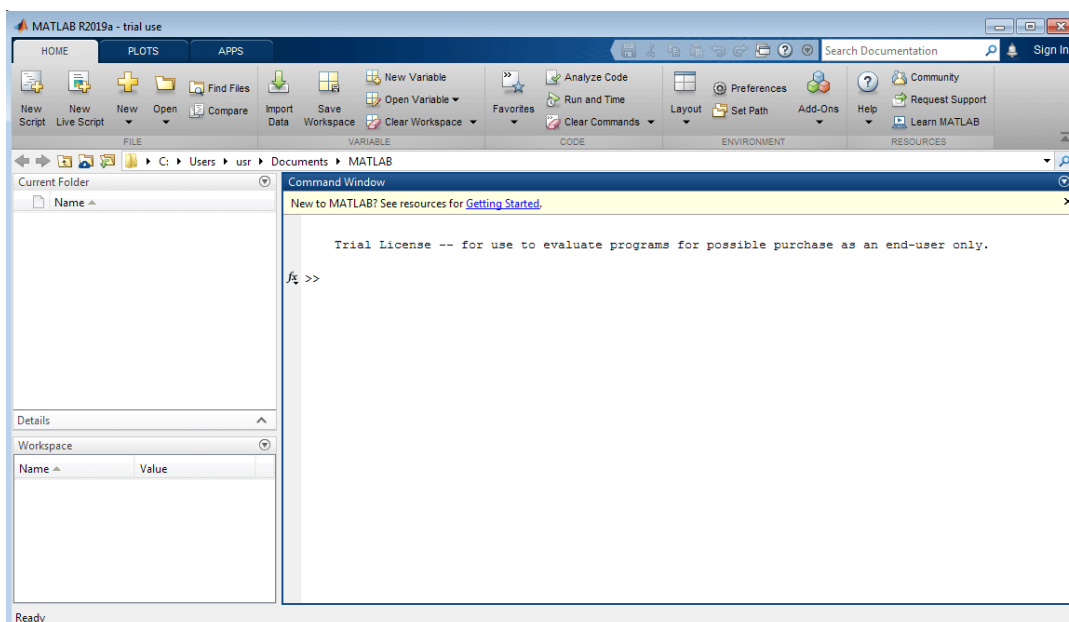


Fig 4.4 – MATLAB Homepage GUI

4.3 THE MATLAB SYSTEM:

The MATLAB system consists of five main parts:

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 "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create complete large and complex application programs.

The MATLAB working environment:

This is the set of tools and facilities that you work with as the MATLAB user or programmer. It includes facilities for managing the variables in your workspace and importing and exporting data. It also includes tools for developing, managing, debugging, and profiling M-files, MATLAB's applications.

Handle Graphics:

This is the MATLAB graphics system. It includes high-level commands for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level commands that allow you to fully customize the appearance of graphics as well as to build complete Graphical User Interfaces on your MATLAB applications.

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 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.

4.3.1 READING IMAGE IN MATLAB:

First, clear the workspace of any variables and close open figure windows.

```
clear all; close all;
```

Read an image into the workspace, using the **imread** command. The example reads one of the sample images included with the toolbox, an image of a cameraman in a file named cameraman.tif , and stores it in an array named I . imread infers from the file that the graphics file format is Tagged Image File Format (TIFF).

```
I = imread('cam.jpg');
```

4.3.2 DISPLAYING IMAGE:

Display the image using imshow function. You can also view an image in the Viewer app. The imtool function opens the Image Viewer app which presents an integrated environment for displaying images and performing some common image processing tasks. The Image Viewer app provides all the image display capabilities of imshow but also provides access to several other tools for navigating and exploring images, such as scroll bars, the Pixel Region tool, Image Information tool, and the Contrast Adjustment tool.

```
imshow(I)
```



Fig 4.5 – Image displayed using imshow() built-in function

4.3.3 WRITING IMAGES

imwrite(A,filename) writes image data A to the file specified by **filename**, inferring the file format from the extension. **imwrite** creates the new file in your current folder. The bit depth of the output image depends on the data type of A and the file format. For most formats:

If A is of data type **uint8**, then **imwrite** outputs 8-bit values.

If A is of data type **uint16** and the output file format supports 16-bit data (JPEG, PNG, and TIFF), then **imwrite** outputs 16-bit values. If the output file format does not support 16-bit data, then **imwrite** returns an error.

If A is a grayscale or RGB color image of data type **double** or **single**, then **imwrite** assumes that the dynamic range is [0,1] and automatically scales the data by 255 before writing it to the file as 8-bit values. If the data in A is **single**, convert A to **double** before writing to a GIF or TIFF file.

If A is of data type **logical**, then **imwrite** assumes that the data is a binary image and writes it to the file with a bit depth of 1, if the format allows it. BMP, PNG, or TIFF formats accept binary images as input arrays.

If A contains indexed image data, you should additionally specify the **map** input argument.

```
z = imread('cam.jpg');
```

```
imwrite(z,'cameraman.png');
```

```
img = imread('cameraman.png');
```

```
imshow(img);
```

4.3.4 TYPECASTING IN MATLAB

`Y = typecast(X, type)` converts a numeric value in `X` to the data type specified by `type`. Input `X` must be a full, noncomplex, numeric scalar or vector. The `type` input is a string set to one of the following: 'uint8', 'int8', 'uint16', 'int16', 'uint32', 'int32', 'uint64', 'int64', 'single', or 'double'.

`typecast` is different from the MATLAB® `cast` function in that it does not alter the input data. `typecast` always returns the same number of bytes in the output `Y` as were in the input `X`. For example, casting the 16-bit integer 1000 to `uint8` with `typecast` returns the full 16 bits in two 8-bit segments (3 and 232) thus keeping its original value ($3 \times 256 + 232 = 1000$). The `cast` function, on the other hand, truncates the input value to 255.

The output of `typecast` can be formatted differently depending on what system you use it on. Some computer systems store data starting with its most significant byte (an ordering called *big-endian*), while others start with the least significant byte

4.4 MATLAB GUI

GUIs (also known as graphical user interfaces or UIs) provide point-and-click control of software applications, eliminating the need to learn a language or type commands in order to run the application.

MATLAB apps are self-contained MATLAB programs with GUI front ends that automate a task or calculation. The GUI typically contains controls such as menus, toolbars, buttons, and sliders. Many MATLAB products, such as Curve Fitting Toolbox, Signal Processing Toolbox™, and Control System Toolbox™ include apps with custom user interfaces. You can also create your own custom apps, including their corresponding UIs, for others to use. Graphical user interfaces (GUIs), also known as apps, provide point-and-click control of your software applications, eliminating the need for others to learn a language or type commands in order to run the application. You can share apps both for use within MATLAB and also as standalone desktop or web apps.

Step 1 : Start GUIDE by typing `guide` at the MATLAB prompt.

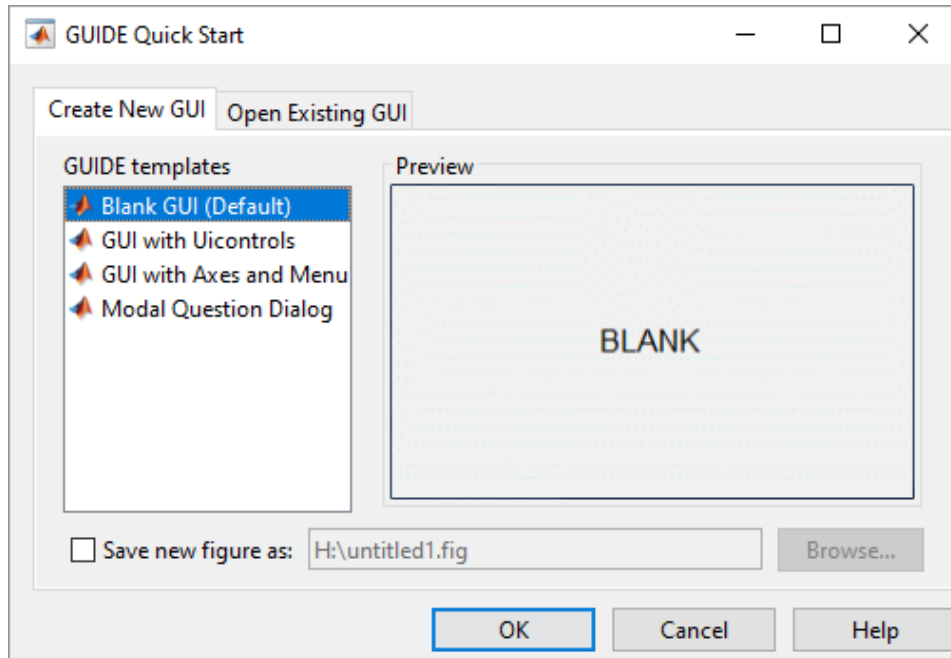


Fig 4.6 – MATLAB Guide prompt

Step 2 : In the GUIDE Quick Start dialog box, select the **Blank GUI (Default)** template, and then click **OK**.

- a. Select **File > Preferences > GUIDE**
- b. Select **Show names in component palette**.
- c. Click **OK**.

Step 3 : Set the size of the window by resizing the grid area in the Layout Editor. Click the lower-right corner and drag it until the canvas is approximately 3 inches high and 4 inches wide. If necessary, make the canvas larger.

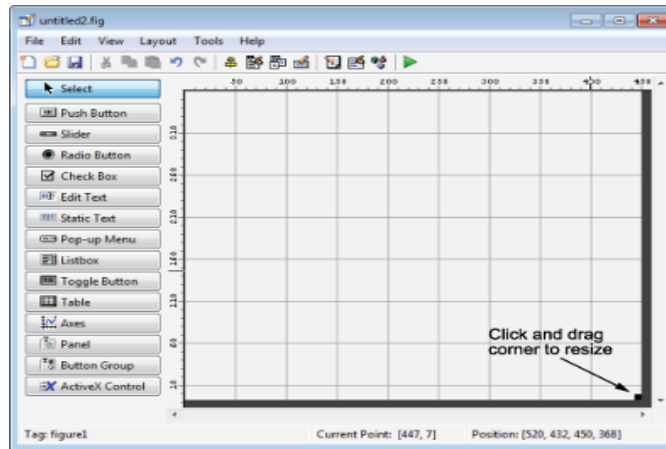


Fig 4.7 – Empty GUI

Step 5: Creating the layout arranging buttons, axes, static text, edit text in order

- **2 Axes** to display the original picture and encrypted picture
- **3 Buttons** to display the original image to encrypt the text and other to decrypt the text
- **1 Edit text** to enter the text which is to be encrypted
- **1 Static text** to display the decrypted text

After arranging all the components, the layout looks like:

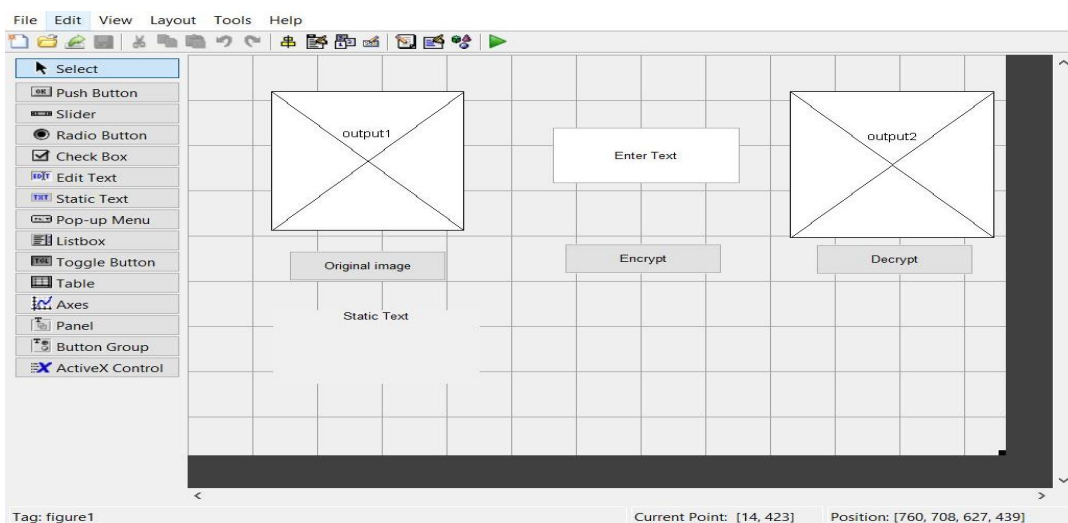


Fig 4.8 – GUI created with MATLAB

After clicking the run button, the following execution window appears



Fig 4.9 – Fully functional GUI

Step 6 : Save the Layout

When you save a layout, GUIDE creates two files, a FIG-file and a code file. The FIG-file, with extension .fig, is a binary file that contains a description of the layout. The code file, with extension .m, contains MATLAB functions that control the app's behavior.

1. Save and run your program by selecting **Tools > Run**.
2. GUIDE saves the files `simple_gui.fig` and `simple_gui.m`, and then runs the program. It also opens the code file in your default editor.

The app opens in a new window. Notice that the window lacks the standard menu bar and toolbar that MATLAB figure windows display. You can add your own menus and toolbar buttons with GUIDE, but by default a GUIDE app includes none of these components. When you run `simple_gui`, you can select a data set in the pop-up menu and click the push buttons, but nothing happens. This is because the code file contains no statements to service the pop-up menu and the buttons.

There are 3 buttons and 2 axes also input and output. Let us go step by step

- By clicking the original image button, the image in which text is going to be encrypted will appear.



Fig 4.10 – Original image

- Next, if we enter some text in the text box and click on the encrypt button then the file text_file.txt is created and the text is saved in that file



Fig 4.11 – Text and image

- The image beside the text box is the encrypted image. The text which is present in the text file is encrypted in the original image and displayed.
- If we press the decrypt button which is below the encrypted image then the text is decrypted from the image and displayed in the static box

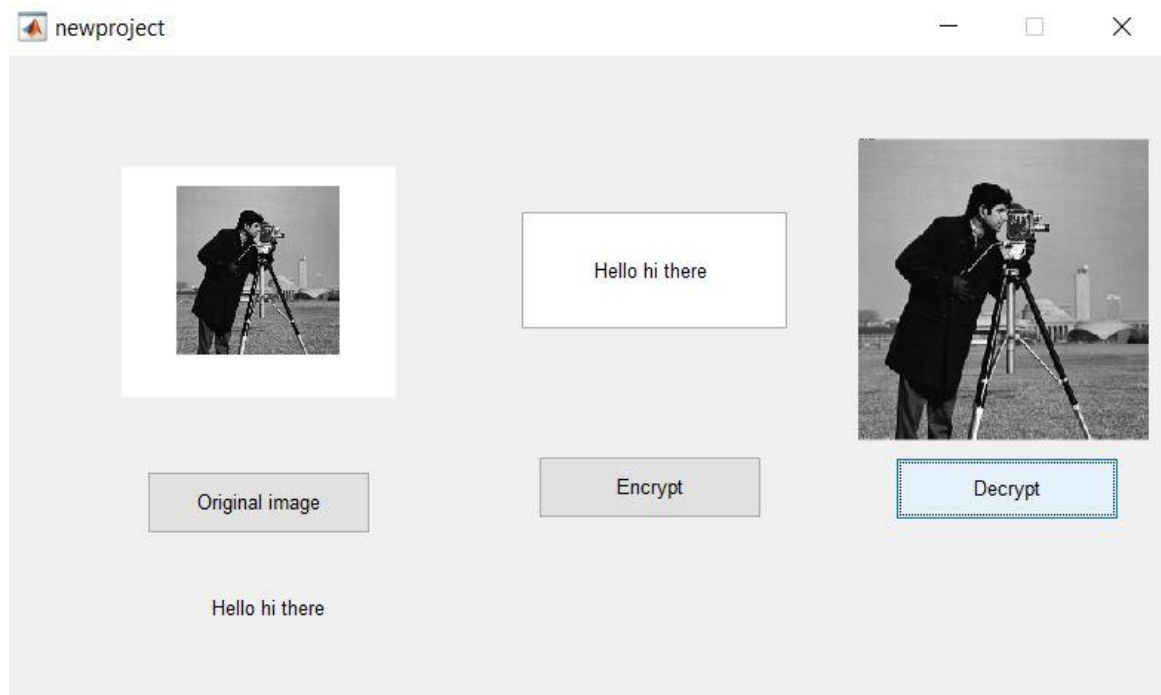


Fig 4.12 – Final GUI

CHAPTER 5

RESULT AND CONCLUSION

In this project we have reviewed on various methods of hiding personal and secure data without being known of its existence. For the solution to arisen problem statement we have created a GUI using a powerful and highly professional software popularly known as MATLAB wherein the project is divided into two parts namely Encryption and Decryption (Refer to the MATLAB codes for encryption and decryption of data over image in the appendix chapter)

MATLAB GUI:

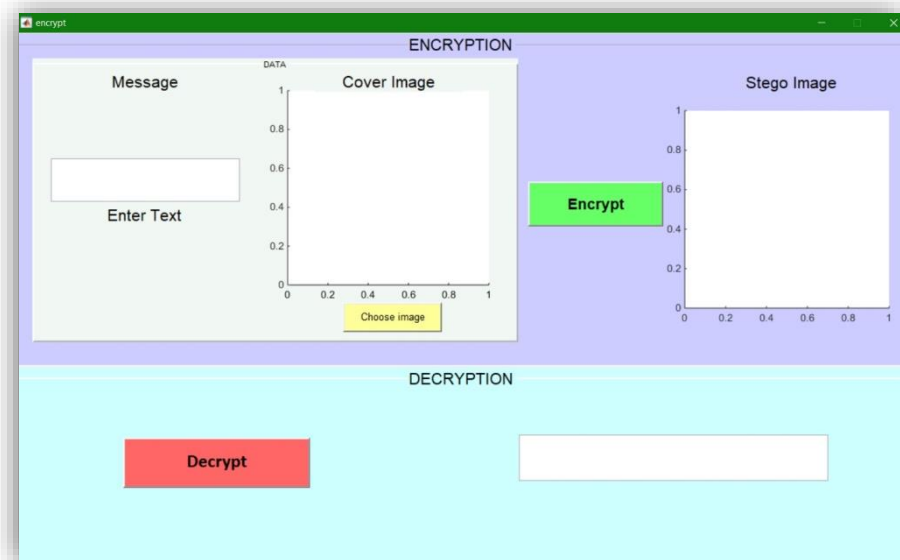


Fig 5.1 – MATLAB GUI

5.1 ENCRYPTION OF DATA:

Secret media encrypted here is a plain text stored in a text file

MESSAGE :



Fig 5.2 - Text file

COVER IMAGE :



Fig 5.3 - Cover image

LSB ALGORITHM :

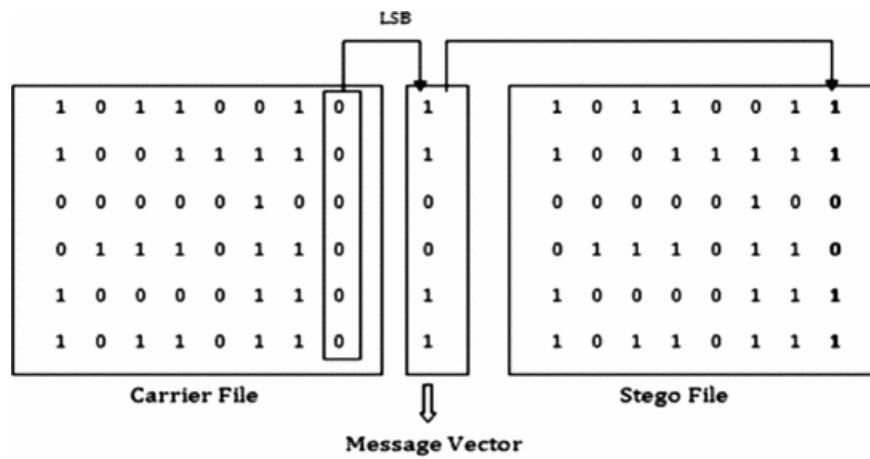


Fig 5.4 - LSB steganography algorithm

ENCRYPTED IMAGE :



Fig 5.5 - Encrypted image

5.2 DECRYPTION OF DATA

DECRYPTED TEXT :



Fig 5.6 – Decrypted text message

The Decrypted text is same as the original text. In our code the original text which is to be encrypted is first converted into numbers according to their ascii values and those ascii values are encrypted in the image. The ascii value is added to the pixel value present in the image this results in the change of colour of the image or noise is added to the image.

Hence the encrypted image appears a bit noisy according to the length of the text. The text is decrypted by performing the subtraction of the pixel values of the encrypted image with the original image. The resultant number which is greater than zero is stored in the array, it is printed to a file in corresponding to its ascii character form of that particular number.

5.3 APPLICATIONS OF IMAGE STEGANOGRAPHY:

There are various applications in steganography; it varies among the user requirements such as copyright control, secret communication, smart ID's, tamper proofing etc.

A. Copyright Control

Inside an image, secret copyright information is embedded. This is achieved by watermarking which is the complex structure so that the intruder cannot identify the copyright information.

There are various methods available to find the watermarking. It is achieved by statistical, correlation, similarity check. Watermarking is used to protect the copyright information.

B. Secret Communication

In many situations, transmitting a cryptographic message draws unwanted attention. The use of cryptographic technology may be restricted or forbidden by law. However, the used steganography does not advertise covert communication and therefore, avoid scrutiny of the sender, message and recipient.

C. Tamper Proofing

The objective of tamper-proofing is to answer the question, “Has this image been modified?” Tamper-proofing techniques are related, but distinct from the other data-hiding technologies. What differentiates them is the degree to which information is secured from the host signal. Most data-hiding techniques attempt to secure data in the face of all modifications. Tamper proofing techniques must be resilient to small modifications (e.g., cropping, tone scale or gamma correction for images or balance or equalization for sounds) but not to large modifications (e.g., removing or inserting people from an image or taking words out of context in an audio recording

D. Digital Watermark

The objective of a digital watermark is to place an indelible mark on an image. Usually, this means encoding only a handful of bits, sometimes as few as one. This “signature” could be used as a means of tracing the distribution of images for an on-line news service and for photographers who are selling their work for digital publication. One could build a digital camera that places a watermark on every photograph it takes. Theoretically, this would allow photographers to employ a “web-searching agent” to locate sites where their photographs appear

E. Smart Id's:

In smart ID's the information about the person is embedded into their image for confidential information. For an organization, the authentication of the resources is accessed by the people. So identifying the theft related to prevention of crimes.

- a. The advantage of steganography, over cryptography alone, is that messages do not attract attention to themselves. Plainly visible encrypted messages—no matter how unbreakable—will arouse suspicion, and may in themselves be incriminating in countries where encryption is illegal. Therefore, whereas cryptography protects the contents of a message, steganography can be said to protect both messages and communicating parties.
- b. This method featured security, capacity, and robustness, the three needed aspects of steganography that makes it useful in hidden exchange of information through text documents and establishing secret communication.
- c. Important files carrying confidential information can be in the server in an encrypted form. No intruder can get any useful information from the original file during transmit.
- d. With the use of Steganography Corporation government and law enforcement agencies can communicate secretly.

5.4 LIMITATIONS:

- a. Huge number of data, huge file size, so someone can suspect about it.
- b. If this technique is gone in the wrong hands like hackers, terrorist, criminals then this can be very much dangerous.

5.5 CONCLUSION

Although only some of the main image steganographic techniques were discussed in this document, one can see that there exists a large selection of approaches to hiding information in images. All the major image file formats have different methods of hiding messages, with different strong and weak points respectively. Where one technique lacks in payload capacity, the other lacks in robustness. Least significant bit (LSB) in both BMP and GIF makes up for this, but both approaches result in suspicious files that increase the probability of detection when in the presence of a warden. The proposed approach in this project uses a new steganographic approach called image steganography. The application creates a stego image in which the personal data is embedded inside the cover file image. We used the Least Significant Bit algorithm in this project for developing the application which is faster and reliable and compression ratio is moderate compared to other algorithms.

By reviewing these papers we observed that most of the steganography work is done in the year 2012-2013. In these years, LSB is the most widely used technique for steganography. Some researchers have also used the techniques like water marking, distortion technique, spatial technique, ISB, MSB in their work and provided a strong means of secure information transmission. These papers provide a lot of help to the initiator for starting their work in this field. This review paper is enough for them to start their work in this field.

5.6 FUTURE SCOPE

The different security and data hiding techniques are used to implement steganography using LSB, ISB, MLSB. In further research we are going to use more advance schemes like steganography with some hybrid cryptographic algorithm for enhancing the data security.

The future work on this project is to improve the compression ratio of the image to the text. This project can be extended to a level such that it can be used for the different types of multimedia files.

REFERENCES

- [1] Rosziati Ibrahim and Teoh Suk Kuan, Steganography Imaging System (SIS): Hiding Secret Message inside an Image
- [2] W.Gonzalez, “Digital Image Processing”, 2nd ed. Prentice Hall, Year of Publication 2008.
- [3] Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, Second Edition.
- [4] Rafael C. Gonzalez, Richard E. Woods, Steve L. Eddins, Digital Image Processing Using MATLAB, 2003.
- [5] Ishwarjot Singh ,J.P Raina,“ Advance Scheme for Secret Data Hiding System using Hop field & LSB” Internation al Journal of Computer Trends and Technology (IJCTT) – volume 4 Issue 7–July 2013.
- [6] G. Manikandan, N. Sairam and M. Kamarasan “A Hybrid Approach for Security Enhancement by Compressed Crypto-Stegno Scheme “, Research Journal of Applied Sciences, Engineering and Technology
- [7] Shabir A. Parah, Javaid A. Sheikh, G.M. Bhat, “Data Hiding in Intermediate Significant Bit Planes, A High Capacity Blind Steganographic Technique”, International Conference on Emerging Trends in Science, Engineering and Technology , pp.192-197, July 2012.
- [8] Michel K. Kulhandjian, Dimitris A. Pados, Ming Li, Stella N. Batalama, and Michael J. Medley, “Extracting spread - spectrum hidden data from digital media “, IEEE transactions on information forensics and security, vol. 8, no. 7, july 2013.
- [9] Chang, Chin-Chen., Lin, Iuan-Chang., and Yaun-Hui YU., “ A new Steganographic method for color and gray scale image hiding”, Computer Vision and Image Understanding, ELSEVIER, Vol. 107, No. 3
- [10] Bailey, K., and Curran, K., “An Evaluation of Image Based Steganography Methods”, Journal of Multimedia Tools and Applications, Vol. 30, No. 1, pp. 55-88, 2006.
- [11] Adnan Gutub, Ayed Al-Qahtani, Abdulaziz Tabakh, “Triple-A: Secure RGB Image Steganography Based on Randomization”, International Conference on Computer Systems and Applications (AICCSA -2009),
- [12] R.Amirtharajan, Sandeep Kumar Behera, Motamarri Abhilash Swarup, Mohamed Ashfaaq and John Bosco Balaguru Rayappan , “Colour Guided Colour Image Steganography” Universal Journal of Computer Science and Engineering Technology , 16-23, Oct. 2010, pp. 2219-2158.
- [13] Anil Kumar , Rohini Sharma,”A Secure Image Steganography Based on RSA Algorithm and Hash -LSB Technique “,International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 7, July 2013.
- [14] A. M. Hamid and M. L. M. Kiah, “Novel Approach for High Secure and High Rate Data Hidden in the Image Using Image Texture Analysis”, International Journal of Engineering and Technology (IJET): 0975 - 4042, (2009)

APPENDIX

MATLAB CODE FOR ENCRYPTION:

```
clear all;

clc; % Clear the command window

input = imread('cam.jpg'); % Read the input image

input=rgb2gray(input); % Convert image to greyscale

input=imresize(input, [512 512]); % Resize the image to required size

message=input('Enter the message to be encoded :'); % Message to be embedded

p=length(message);

len = length(message) * 8; % Length of the message where each character is 8 bits

ascii_value = uint8(message); % Get all the ASCII values of the characters of the message

bin_message = transpose(dec2bin(ascii_value, 8)); % Convert the decimal values to binary

bin_message = bin_message(:);% Get all the binary digits in separate row

N = length(bin_message); % Length of the binary message

bin_num_message=str2num(bin_message); % Converting the char array to numeric array

output = input; % Initialize output as input

height = size(input, 1); % Get height and width for traversing through the image

width = size(input, 2);

embed_counter = 1; % Counter for number of embedded bits

for i = 1 : height % Traverse through the image
```

```

for j = 1 : width

    if(embed_counter <= len) % If more bits are remaining to embed

        LSB = mod(double(input(i, j)), 2); % Finding the Least Significant Bit of the current

        temp = double(xor(LSB, bin_num_message(embed_counter))); % Find whether the
bit is same or needs to change

        output(i, j) = input(i, j)+temp; % Updating the output to input + temp

        embed_counter = embed_counter+1; % Increment the embed counter

    end

end

end

% Write both the input and output images to local storage

% Mention the path to a folder here.

output(512,512)=p

filename = 'output_img.xlsx';

xlswrite(filename, output);

imwrite(input, 'originalImage.png');

imwrite(output, 'stegoImage.png');

```

DECRYPTION CODE

```
clear all; % Clear the existing workspace

clc; % Clear the command window

filename = 'output_img.xlsx'; % Getting the input image

input_image = xlsread(filename);

height = size(input_image, 1); % Get height and width for traversing through the image

width = size(input_image, 2);

chars = input_image(512,512); % Number of characters of the hidden text

message_length = chars * 8; % Number of bits in the message

counter = 1; % counter to keep track of number of bits extracted

for i = 1 : height % Traverse through the image

    for j = 1 : width

        if (counter <= message_length) % If more bits remain to be extracted

            extracted_bits(counter, 1) = mod(double(input_image(i, j)), 2); % Store the LSB of
the pixel in extracted_bits

            counter = counter + 1; % Increment the counter

        end

    end

end

binValues = [ 128 64 32 16 8 4 2 1 ]; % Powers of 2 to get the ASCII value from binary

% Get all the bits in 8 columned table
```

```
% Each row is the bits of the character  
  
% in the hidden text  
  
binMatrix = reshape(extracted_bits, 8,(message_length/8));  
  
% Convert the extracted bits to characters  
  
% by multiplying with powers of 2  
  
textString = char(binValues*binMatrix);  
  
  
  
disp(textString); % Print the hidden text
```