

ABSTRACT:

The rapid development of data transfer through internet has made it easier to send the data accurate and faster to the receiver end. Many transmission media are available in today's world to transfer the data to destination like e-mails, social sites etc. Many approaches like cryptography are used to transfer the data securely to the destination without any modifications.

For this project we explore two methods for encoding/storing a message within the RGB pixels of a cover image without visual distortion. For both methods a message is broken down to individual components, converted into 8-bit binary values, encrypted using a simple symmetric Exclusive OR (XOR) encryption key, and then encoded in the cover image by changing the least significant bit of the pixel value. The first method implemented sequentially stores the message starting with the top left pixel and then encodes the message from top to bottom and left to right. The second method implemented randomly encodes pixels across the entire image in an attempt to be less noticeable to analysis.

Keywords— Information Hiding, Multimedia Messaging Service, Smart Phones, Steganalysis, Cryptography, Internet Security, Security Attacks, Elliptic Curve Cryptography.

CHAPTER 1

1. INTRODUCTION

In the current trends of the world, the technologies have advanced so much that most of the individuals prefer using the internet as the primary medium to transfer data from one end to another across the world. There are many possible ways to transmit data using the internet: via e-mails, chats, etc. The data transition is made very simple, fast and accurate using the internet. However, one of the main problems with sending data over the internet is the “security threat” it poses i.e. the personal or confidential data can be stolen or hacked in many ways. Therefore it becomes very important to take data security into consideration, as it is one of the most essential factors that need attention during the process of data transferring.

Data security basically means protection of data from unauthorised users or hackers and providing high security to prevent data modification. This area of data security has gained more attention over the recent period of time due to the massive increase in data transfer rate over the internet.

In order to improve the security features in data transfers over the internet, many techniques have been developed like: Cryptography, Steganography and digital watermarking. While Cryptography is a method to conceal information by encrypting it to “cipher texts” and transmitting it to the intended receiver using an unknown key,

Steganography provides further security by hiding the cipher text into a seemingly invisible image or other formats.

According to Johnson et al., (2001), “Steganography is the art of hiding and transmitting data through apparently innocuous carriers to conceal the existence of data”. The level of visibility is decreased using many hiding techniques in “Image Modelling” like “LSB Manipulation”, “Masking and filtering”. These techniques are performed by different steganographic algorithms like F5, LSB, JSteg etc. and the act of detecting the information hidden through these algorithms is called “Steganalysis”. “Cryptography” is the art of science used to achieve security by encoding the data to transform them into non readable formats so that unauthorized users cannot gain access to it.

The encoded text is known as “Cipher text” and this technique is known as encryption and this

process is reversed with authorised access using the decryption technique, in which the encoded data is decoded into readable format.

“Steganography” and “Cryptography” are closely related constructs. The hidden or embedded image, audio or a video files act as carriers to send the private messages to the destination without any security breach. Steganography techniques can be implemented on various file formats such as audio (“.mp3”, “.wmv.”, etc.), video (“.mpeg”, “.dat”, etc.) and images (“.jpeg”, “.bmp”, etc.). However, the images are the most preferred file format for this technique. At present, there are a lot of algorithms that help in executing the steganography software. These tools are.

“Digital watermarking” is described as one of the possibilities to close the gap between copyright issues and digital distribution of data. It is mainly based on Steganographic techniques and enables useful safety mechanisms .

It acts as a very good medium for copyright issues as it embeds a symbol or a logo in the form of a watermark, which cannot be altered manually.

One critical factor to be kept in mind when using steganography is to prevent any further alterations to the originality of the image after embedding the data. Whenever the image with the secret data is transmitted over the internet unauthorised parties may want to hack the data hidden over the image. So, if the originality of the image has been changed then it will be easier to hack the information by unauthorised persons. In order to improve the security, the Digital watermarks are predominantly inserted as transformed digital signal into the source data using key based embedding algorithm and pseudo noise pattern.

This technique has also found big use in the notorious hands of terrorists and the September 2001 Twin tower attacks of the USA are predominantly associated with the communications using steganography. The Steganalysis aims at discovering and decrypting the suspected data transferred with the use of the available algorithms.

1.1 OBJECTIVE OF THE PROJECT:

The aim of the project is to encrypt the data i.e., hide the data over an image using different steganographic algorithms and to compare those algorithms in the context of speed, quality of concealing and the use of watermarks and to describe their functionality in data security.

1.2 PROPOSED METHOD:

In this project, we use a method of encrypting the text and audio files in an image file in order to test the accuracy and efficiency of encryption. This process helps to send the information to the authorised party without any potential risk. The proposed method will help to secure the content with in the image and encryption of audio file with in the image will help to make the document much securer because even though if the unauthorised person succeeds in being able to hack the image, the person will not able to read the message as well as acquire the information in the audio file.

In this research, we will compare three steganographic algorithms in order to compare the hiding capacity and efficiency of hiding the message with in an image. Whenever the audio or data is encrypted using steganographic algorithms with in image, neither the audio/data nor the image it is embedded in should lose its originality. Hence, we compare the different algorithms used for steganography for the various hiding techniques and formats and analyse the results obtain

The process consists of

- Providing security for the data to be transmitted through network using steganography.
- Using digital watermarking techniques
- Implementing different steganographic algorithms
- Comparing different steganographic algorithms in means of speed, accuracy and quality of hiding.
- Proposing an approach for hiding the data within an image using a steganographic algorithm which provides better accuracy and quality of hiding.

The Matlab software 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. The file's changes in resolution due to the pixels lost are analyzed for suggesting the optimal method for the technique.

1.3 OVERVIEW OF THE PROJECT:

In my project we primarily concentrated on the data security issues when sending the data over the network using steganographic techniques. The main objectives of the project are

- Overview of different steganographic algorithms and comparing them in means of speed and quality of hiding.
- Testing the efficiency and accuracy of hiding the data through algorithms using different software.

1.4 SCOPE AND LIMITATIONS:

The scope of the project is to limit unauthorised access and provide better security during message transmission. To meet the requirements, we use the simple and basic approach of steganography and digital watermarking. In this project, the proposed approach finds the suitable algorithm for embedding the data in an image using steganography which provides the better security pattern for sending messages through a network.

For practically implementing the function of the discussed algorithms, Matlab framework is used. Although the Matlab is not particularly known for its top security functionalities, we use this for easier application development and a well-defined User Interface.

1.5 ORGANIZATION OF THESIS:

Chapter-1: Introduction: In this section, the main points discussed are about the Overview, the Background of the project, the scopes and limitations of the project and the approach to research employed are discussed.

Chapter-2: Literature Review: Definitions and overview about the different information security methods to gather knowledge on the existing theories of steganography and review it for proposing an improvised system for providing the required security and discuss about different functionalities of algorithms used for the proposed system.

Chapter-3: Design Structure: This section describes the general architecture of encryption, decryption and data hiding procedures using Data Flow Diagrams.

Chapter-4: Implementation: Description about the software requirements for the proposed system, overview of the Matlab software and implementations of different modules like encryption, decryption and data hiding techniques. It also discusses about the advantages of the Matlab system over the other frameworks.

Chapter-5: Testing: Here, the algorithm proposed to analyse in different formats and analyse on its operations is tested and error reports are prepared. The different types of testing helps are considered to validate the built software on different conditions.

Chapter-6: Conclusion and Future work: Here, the project is concluded with the results of the proposed method that has been analysed and recommendations are made according to the results obtained from the analysis.

CHAPTER 2

2. LITERATURE REVIEW

2.1 INFORMATION SECURITY

In general, security denotes “the quality or state of being secure to be free from danger” (Whitman, 2007, pp.09). Security is classified into different layers depending on the type of content intended to be secured:

Physical security: Defines the required issues that are needed to protect the physical data or objects from unauthorized intrusion.

Personal security: It is defined as the security of the individuals who are officially authorized to access information about the company and its operations

Operational security: It mainly relies on the protection of the information of a particular operation of the chain of activities.

Communication’s security: The communication’s security encompasses the security issues regarding the organization’s communication media, technology and content.

Network security: The network security is responsible for safeguarding the information regarding the „networking components“, „connections“ and contents.

Information security:

Information security is the protection of information and the systems and hardware that use, store, and transmit that information. Information security can be defined as measures adopted to prevent the unauthorized use or modification of use of data or capabilities.

The main objective of the project is to propose the method and critically discuss the properties which help to transmit the data or information over a network without any modifications. The critical characteristics of information are

1. Availability
2. Accuracy

3. Authenticity
4. Confidentiality
5. Integrity

Availability: prevention of unauthorised disclosure of information. It enables users who need access the information to do so without any interference or obstruction and to receive it in the required format. The availability of information requires the verification of the user as one with authorized access to information (**Whitman,2007**).

In other words the availability can be defined as “Ensuring timely and reliable access to make use of information. A loss of availability is the disruption of access to or use of information or an information system” (**Stallings, 2007, pp.09**).

Accuracy: The information is deemed accurate if it does not contain any mistakes / errors and possesses the value that end user expects. If the information holds a value different from that of the end user’s expectations because of intentional or unintentional modifications of its content it becomes no longer accurate (**Whitman,2007**).

Authenticity: Authenticity refers to the quality or state of being genuine or original. It should not be a reproduction or fabrication of any previously known data. The Information is considered authentic when it is originally created, placed, stored or transferred. In general, authenticity is ensuring that all the data remains in its original state by stopping any ways of the unauthorized modification of information (**Whitman, 2007**).

Confidentiality: “The confidentiality is the quality or state of preventing disclosure or exposure to unauthorized individuals or system”. Confidentiality is basically privacy and secrecy which means protection of personal data or that of data belonging to an organisation. Confidentiality of information ensures that only those with the rights and privileges access a particular set of information and prevent from unauthorized access (**Whitman, 2007**).

Integrity: It is the prevention of unauthenticated modification of data. “The quality or state of being whole, complete and uncorrupted is the integrity of information”. The integrity of any data is lost when it is subjected to corruption, damage (external / internal), destruction or other disruption of its authentic state by intended or unintended sources (**Whitman, 2007**).

2.1.1 Security attacks:

The data is transmitted from source to destination which is known as its normal flow as shown in the figure. But the hackers might hack the network in order to access or modify the original data. These types of attacks are formally known as security attacks.

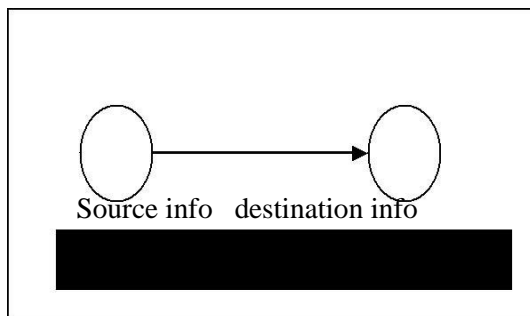


Figure 1: Normal data flow

A hacker can disrupt this normal flow by implementing the different types of techniques over the data and network in following ways. They are:

- Interruption
- Interception
- Modification
- Fabrication

- **Interruption:**

Interruption is an attack by which the hackers can interrupt the data before reaching the destination. This type of attack shows the effect on availability and usually destroys the system asset and makes the data unavailable or useless.

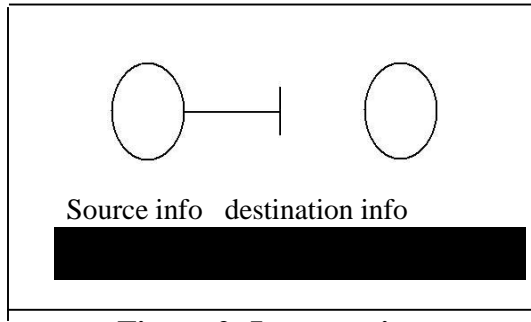


Figure 2: Interruption

Interception:

Interception is one of the well known attacks. When the network is shared that is through a local area network is connected to Wireless LAN or Ethernet it can receive a copy of packets intended for other device. On the internet, the determined hacker can gain access to email traffic and other data transfers. This type of attack shows the effect on confidentiality of data.

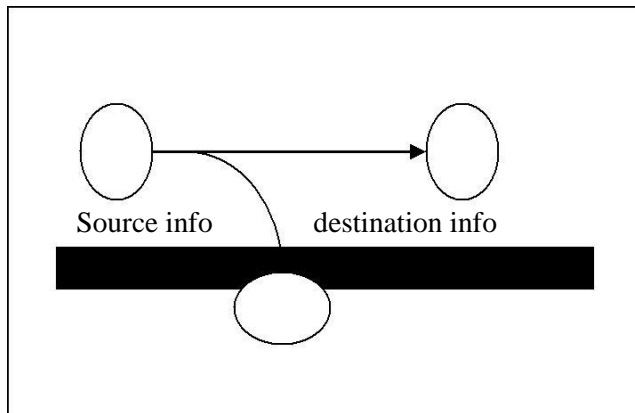


Figure 3: Interception

Modification:

This refers to altering or replacing of valid data that is needed to send to destination. This type of attacks is done usually by unauthorized access through tampering the data. It shows effect on the integrity of the data.

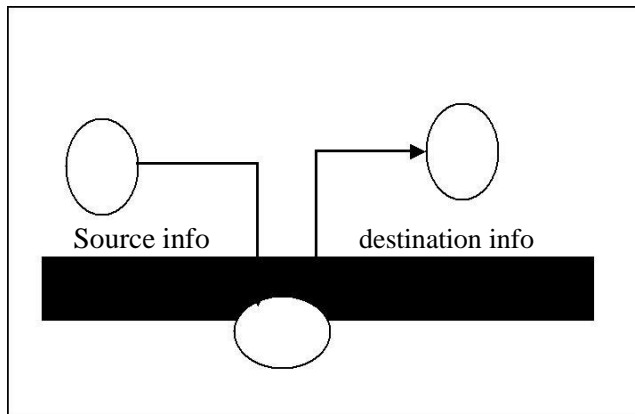


Figure 4: Modification

Fabrication:

In this type, the unauthorized user places data without the interface of source code. The hacker or unauthorized person inserts the unauthorized objects by adding records to the file, insertion of spam messages etc. This type of attack affects on the Authenticity of message.

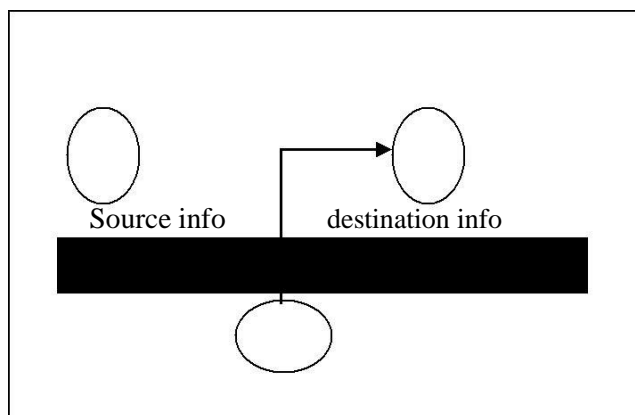


Figure 5: Fabrication

There are many types of security attacks that will try to modify the original data. The main goal of any organisation / individual transmitting the data is to implement security measures which include

1. Prevention
2. Detection
3. Response
4. Recovery

Prevention: The security attacks can be prevented by using an encryption algorithm to restrict any unauthorized access to the encryption keys. Then the attacks on confidentiality of the transmitted data will be prevented.

Detection: Using the intrusion detection systems for detection of unauthorized individuals logged onto a system and making the resources available to legitimate users.

Response: Whenever the unauthorised attacks happen in the system, the security mechanisms can detect the process and the system can respond to make the data unavailable.

Recovery: Recovery is the final approach if an attacker modifies the data or makes the data unavailable. The data can then be recovered by using backup systems, so that the integrity of the data shall not be compromised.

There are different types of approaches for preventing the security attacks. The most useful approaches are

1. Cryptography
2. Steganography
3. Digital watermarking

2.2 CRYPTOGRAPHY

The word cryptography is derived from two Greek words which mean “secret writing”.

Cryptography is the process of scrambling the original text by rearranging and substituting the original text, arranging it in a seemingly unreadable format for others. Cryptography is an effective way to protect the information that is transmitting through the network communication paths (**Bishop, 2005**).

Cryptology is the science that deals about cryptography and cryptanalysis. Cryptography is the approach of sending the messages secretly and securely to the destination. Cryptanalysis is the

method of obtaining the embedded messages into original texts (Whitman, 2007).

In general, cryptography is transferring data from source to destination by altering it through a secret code. The cryptosystems uses a plaintext as an input and generate a cipher text using encryption algorithm taking secret key as input.

The important elements in cryptosystems are

1. Plain text (input)
2. Encryption algorithm
3. Secret key
4. Cipher text
5. Decryption algorithm

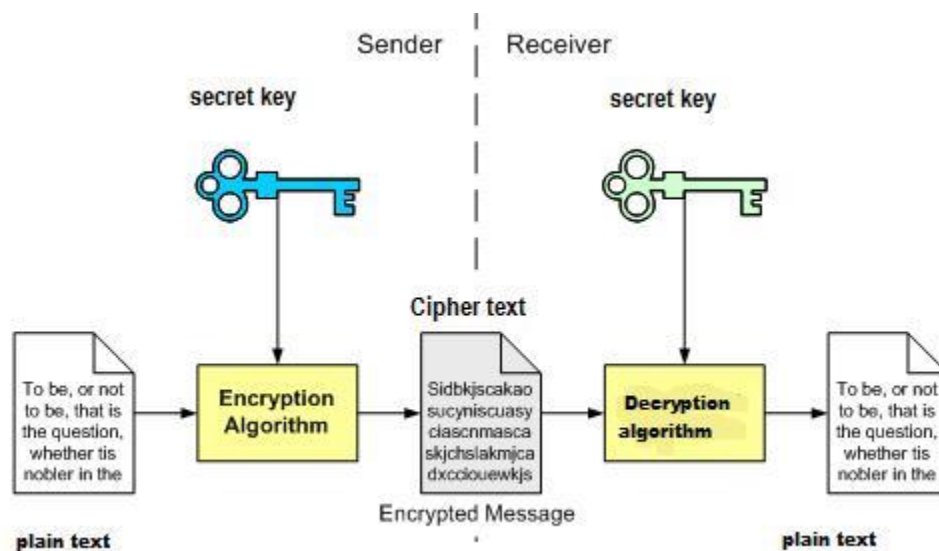


Figure 6: General model of cryptographic system

Plain text: The plain text is an original piece of information that is needed to send information to the destination.

Encryption algorithm: This is the main key to any cryptographic system. This encryption algorithm subjects the plain text to various substitutions and transformations.

Secret key: The secret key is given by the user which will act as an input to the encryption algorithm. Based on this key, various substitutions and transformations on the plain text will differ.

Cipher text: This is the output generated by the encryption algorithm. The cipher text is the jumbled text. The cipher text differs with each and every secret key that has given to the encryption algorithm.

Decryption algorithm: This is opposite to the „encryption algorithm“. It will acquire cipher text and secret key as an input and produce plain text as an output.

Cryptographic Algorithms: There are many cryptographic algorithms available which differ on their type of encryption. Based on the type of encryption standards the algorithms are grouped into two types

1. Symmetric encryption algorithm
2. Asymmetric encryption algorithm

2.2.1 Symmetric Encryption

Symmetric encryption is a single key encryption and also known as conventional encryption. It is also referred as „private key cryptography“. The symmetric encryption algorithm generally uses the same key for „encryption“ and „decryption“. The security level for this type of encryption will depend on the length of the key.

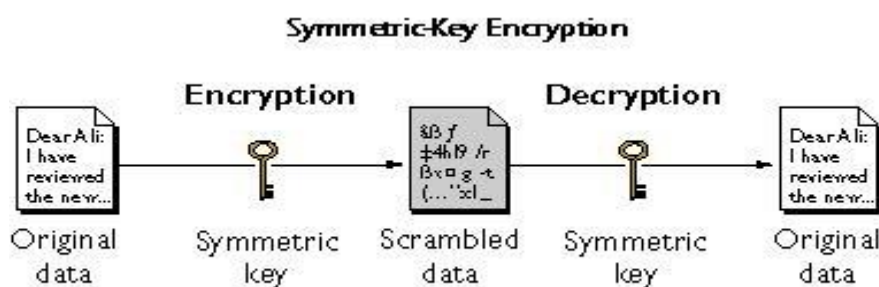


Figure 7: Symmetric encryption

There are two types of methods that will attack on symmetric encryption systems. The first one is

Cryptanalysis. If the attacker gets to know some information about the plain text and cipher text, he analyses the characteristics of the algorithms used for encryption and tries to generate keys. The second type of attack is known as

„brute force attack“. In this type of attack, the defender attempts to know the cipher text and try every possible key for translation. To avoid this problem, the user should use the key that no longer can be estimated like 128 or 168 bit keys (**Alfred J, M et al., 1996**).

Block ciphers: Block cipher is an asymmetric algorithm in which the cipher processes the text in fixed size blocks and generates same size cipher text blocks. In this algorithm, the plaintext is divided into independent blocks of 8-16 bytes and encrypts each block independently.

The different symmetric encryption algorithms are

- Data encryption standard
- Advanced encryption standard

Data encryption standard (DES):

“Data Encryption Standard” (DES) is also known as Data Encryption Algorithm (DEA). DEA takes 64 bits of plain text and 56 bits of key to produce 64 bits cipher text block. The DES algorithm always functions on blocks of equal size and uses the permutations and substitutions in algorithm.

The data encryption algorithm uses 56 bit key so it is not possible for the defender for analysing the key. So, the problem of Cryptanalysis is avoided using this algorithm. But the drawback of the algorithm is Brute-force attack. This can be avoided using the Triple DES algorithm.

Triple DES:

Triple DES is an extension to the DES algorithm. Triple DES uses the same approach for encryption as DES. 3DES takes three 64 bit keys which has a total length of 192 bits. We can give more than one key that is two or three keys for encryption as well as for decryption such that the security will be stronger. It is

approximately 2^{56} times stronger than the normal DES algorithm, so that this algorithm can avoid the brute force attack. The main drawback of using 3DES algorithm is that the number of calculations is high reducing the speed to a greater extent. And the second drawback is that both DES and 3DES use same 64 block size to avoid security issues. “Advanced Encryption Standard” algorithms are used to avoid these limitations.

Advanced Encryption Standards:

Advanced Encryption Standards (AES) takes a block of size 128 bits as input and produces the output block of same size. AES supports different key sizes like 128, 192 and 256 bit keys. Each encryption key size will change the number of bits and also the complexity of cipher text.

The major limitation of AES is error propagation. The encryption operation and key generation both engage in number of non linear operations, so, for lengthy operations it is not suitable. A cryptanalyst may be able to use the continuities in plain text to simplify the decryption (**Whitman, 2007**).

2.2.2 Asymmetric Encryption

„Asymmetric encryption“ is also known as „Public key encryption“. The AES works same as Symmetric encryption, the main difference between AES and Symmetric encryption is in using keys. In asymmetric encryption, the encryption and decryption will be done by two different keys. It will use plain text, encryption algorithm and decryption algorithm same as Symmetric encryption as discussed in above section.

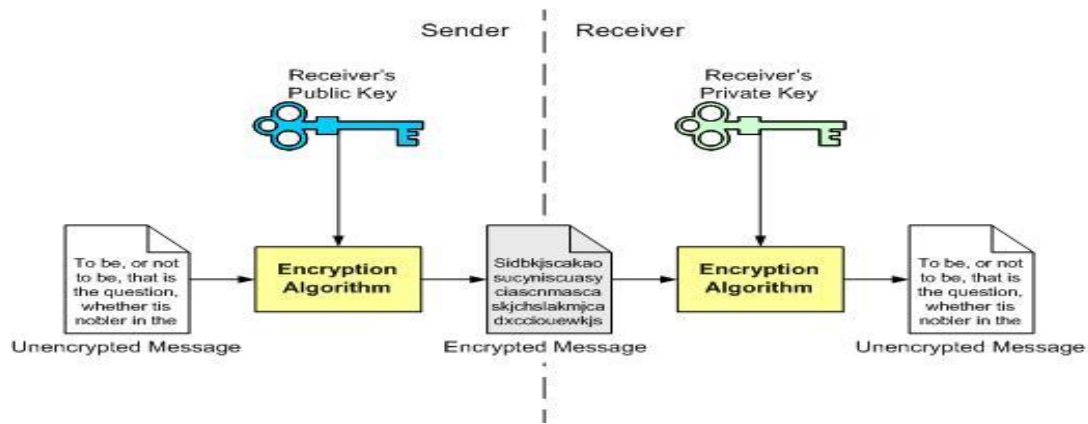


Figure 8: Asymmetric Encryption

In „Asymmetric encryption“, only the data that is encrypted using public key can be decrypted using the same algorithm. And the message which is encrypted using private key can be decrypted using only the matching public key.

The main problem with Asymmetric algorithm is “cipher keys”. Whenever two different people want to exchange the data simultaneously using asymmetric encryption they need to have four different keys. It will be more confusing to resolve as the corresponding key is required for the particular file to open.

The most important public key encryption algorithm is RSA algorithm

RSA:

RSA was first developed in 1977. RSA functions depend upon the large prime numbers of public and private keys. The security is also based on the difficulty of prime numbers. The RSA algorithms are used in public key encryptions as well as in digital signatures. It allows the sender to encrypt the message using public key and decrypt the message using private key by receiver. So, the security will be high using RSA in public key encryption (Stallings, 2007).

2.3 STEGANOGRAPHY

Steganography in Greek means „covered writing“. Steganography is the process of hiding the one information into other sources of information like text, image or audio file, so that it is not visible to the natural view. There are varieties of steganographic techniques available to hide the data depending upon the carriers we use.

Steganography and cryptography both are used for the purpose of sending the data securely. The same approach is followed in Steganography as in cryptography like encryption, decryption and secret key. In steganography the message is kept secret without any changes but in cryptography the original content of the message is differed in different stages like encryption and decryption.

Steganography supports different types of digital formats that are used for hiding the data. These files are known as carriers. Depending upon the redundancy of the object the suitable formats are used. „Redundancy“ is the process of providing better accuracy for the object that is used for display by the bits of object.

The main file formats that are used for steganography are Text, images, audio, video, protocol (Morkel, 2005).

The different types of steganographic techniques that is available are

1. Pure steganography
2. Public key steganography
3. Secret key steganography

Pure steganography: Pure steganography is the process of embedding the data into the object without using any private keys. This type of steganography entirely depends upon the secrecy. This type of steganography uses a cover image in which data is to be embedded, personal information to be transmitted, and encryption decryption algorithms to embed the message into image.

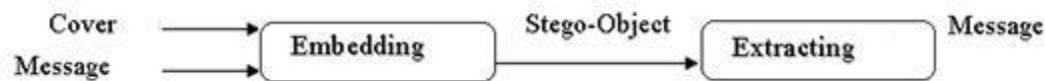


Figure 9: pure steganography process (Zaidoon, 2010).

This type of steganography can't provide the better security because it is easy for extracting the message if the unauthorised person knows the embedding method. It has one advantage that it reduces the difficulty in key sharing (Zaidoon, 2010).

Secret key steganography: Secret key steganography is another process of steganography which uses the same procedure other than using secure keys. It uses the individual key for embedding the data into the object which is similar to symmetric key. For decryption it uses the same key which is used for encryption.

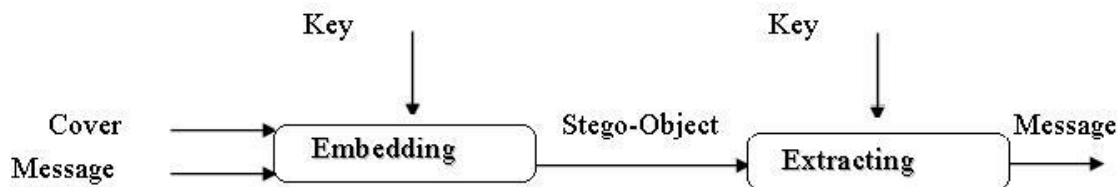


Figure 10: secret key steganography (Zaidoon, 2010).

This type of steganography provides better security compared to pure steganography. The main problem of using this type of steganographic system is sharing the secret key. If the attacker knows the key it will be easier to decrypt and access original information (Zaidoon, 2010).

Public key steganography: Public key steganography uses two types of keys: one for encryption and another for decryption. The key used for encryption is a private key and for decryption, it is a „public key“ and is stored in a public database (Zaidoon, 2010).

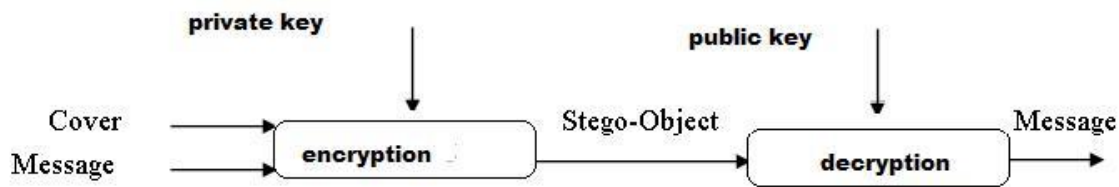


Figure 11: public key steganography (Zaidoon, 2010).

For encryption and decryption of text messages using the secret keys steganographic system uses algorithms known as steganographic algorithms. The mostly used algorithms for embedding data into images are

1. LSB (Least Significant Bit) Algorithm
2. JSteg Algorithm
3. F5 Algorithm

2.3.1 LSB algorithm

LSB (Least Significant Bit) substitution is the process of adjusting the least significant bit pixels of the carrier image. It is a simple approach for embedding message into the image. The Least Significant Bit insertion varies according to number of bits in an image. For an 8 bit image, the least significant bit i.e., the 8th bit of each byte of the image is changed to the bit of secret message. For 24 bit image, the colours of each component like RGB (red, green and blue) are changed. LSB is effective in using BMP images since the compression in BMP is lossless. But for hiding the secret message inside an image of BMP file using LSB algorithm it requires a large image which is used as a cover. LSB substitution is also possible for GIF formats, but the problem with the GIF image is whenever the least significant bit is changed the whole colour palette will be changed. The problem can be avoided by only using the gray scale GIF images since the gray scale image contains 256 shades and the changes will be done gradually so that it will be very hard to detect. For JPEG, the direct substitution of steganographic techniques is not possible since it will use lossy compression. So it uses LSB substitution for embedding the data into images. There are many

approaches available for hiding the data within an image: one of the simple least significant bit submission approaches is „Optimum Pixel Adjustment Procedure“. The simple algorithm for OPA explains the procedure of hiding the sample text in an image.

Step1: A few least significant bits (LSB) are substituted with in data to be hidden.

Step2: The pixels are arranged in a manner of placing the hidden bits before the pixel of each cover image to minimize the errors.

Step3: Let n LSBs be substituted in each pixel.

Step4: Let d= decimal value of the pixel after the substitution. $d1 =$
decimal value of last n bits of the pixel.

$d2 =$ decimal value of n bits hidden in that pixel. Step5:

If $(d1 \sim d2) \leq (2^n)/2$

then no adjustment is made in that pixel. Else

Step6: If $(d1 < d2)$

$d = d - 2^n.$

If $(d1 > d2)$

$d = d + 2^n.$

This, d'' is converted to binary and written back to pixel (Amirtharajan et al., 2010).

This method of substitution is simple and easy to retrieve the data and the image quality better so that it provides good security.

2.3.2 JSTEG algorithm

JSteg algorithm is one of the steganographic techniques for embedding data into JPEG images. The hiding process will be done by replacing Least Significant Bits (LSB). JSteg algorithm replaces LSBs of quantized Discrete Fourier Transform (DCT) coefficients. In this process the hiding mechanism skips all coefficients with the values of 0 or 1. This algorithm is resistant to

visual attacks and offers an admirable capacity for steganographic messages. Generally, JSteg steganographic algorithm embedded the messages in lossy compressed JPEG images. It has high capacity and had a compression ratio of 12%. JSteg algorithm is restricted for visual attacks and it is less immune for statistical attacks. Normally, JSteg embeds only in JPEG images. In these JPEG images, the content of the image is transformed into

“frequency coefficients” so as to achieve storage in a very compressed format. There is no visual attack in the sense presented here, due to the influence of one steganographic bit up to 256 pixels (Ahmed et al., 2006).

2.3.3 F5 algorithm

F5 algorithm was introduced by German researchers Pfitzmann and Westfeld in order to avoid the security problem when embedding the data into the JPEG images. The F5 algorithm embeds the message into randomly chosen Discrete Fourier Transform (DCT) coefficients. It utilizes matrix embedding which minimises the changes to be made to the length of certain message. The F5 Algorithm provides high steganographic capacity, and can prevent visual attacks. F5 algorithm is also resistant to statistical attacks. This algorithm uses matrix encoding such that it reduces the number of changes needed to embed a message of certain length. This algorithm avoids the chi-square attack since it doesn’t replace or exchange the bits.

The resistance is high for both visual and statistical attacks. It has high embedding capacity that is greater than 13%. This algorithm supports TIFF, BMP, JPEG and GIF formats (Cox et al., 2003).

The performance of the algorithms differs with the type of cover image or source on which the data is embedded. The comparison of algorithms is tabulated below.

Steganographic algorithm	Speed	Quality of hiding	Security
LSB	High	Good	Less
F5	High	High up to 13.4%	High & Strong
JSteg	Moderate	Embedding capacity up to 12%	Less

Table 1: Comparison of different Steganographic Algorithms

2.4 DIGITAL WATERMARKING

“Watermarking is the practice of imperceptibly altering work to embed a secret message” (Miller et al., 2008).

„Digital watermarking“ is the process of inserting information into a digital signal. The main aim of digital watermarking is to protect the integrity and authenticity of digital media. Digital watermarking directly embeds a watermark containing owner identification into the host signal in such a way that the hacker can’t remove the watermark without reducing the quality of the signal or an image. Digital watermarks can be used as proof of authorization and can be used as a signature which shows the ownership of particular asset like images, video and audio files.

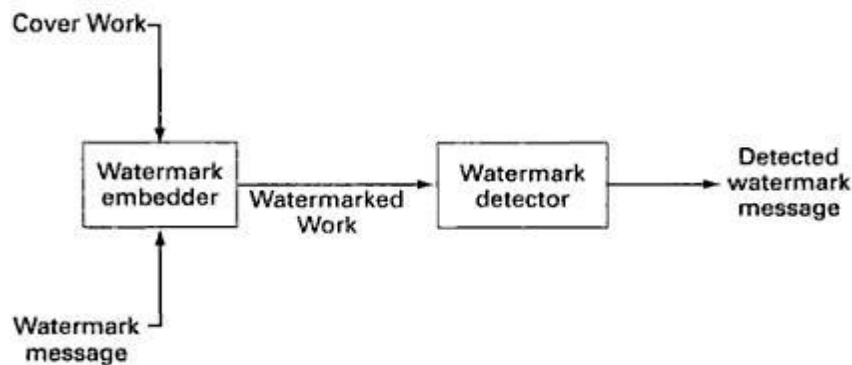


Fig 12: General watermarking system (Cox et al., 2008, pp.03).

There are two type of watermarking techniques one is robust watermarking and another is fragile watermarking. Robust watermarking is mainly used for the purpose of copyright protection because they are strong for all kinds of manipulations in images. The second method fragile watermarking is used for providing better authentication and for verification of integrity in order to avoid the modifications

(Yang et al., 2010).

The applications of watermarking are:

- Copyright protection: Watermarks are used for copyright protection by embedding the watermark secretly which can be read only through the secret key held by the owner.
- Monitoring: Watermarks are used for tracing the illegal copying.
- Finger printing: In the „point to point distribution“ environments, the information on the authenticated customers could be embedded into secret watermarks well before the secure delivery of the data.
- „Content manipulation indication“: The indication of content manipulation from the authorised state can be detected only by means of a public or fragile watermark.
- „Information carrier“: A „public watermark“ is embedded into the data stream that shall act as a link to the external databases to store information about the copyright and license conditions **(Arnold, 2000).**

CHAPTER 3

3. DESIGN

The data hiding patterns using the steganographic technique in this project can be explained using this simple block diagram. The block diagram for steganographic technique is as follows.

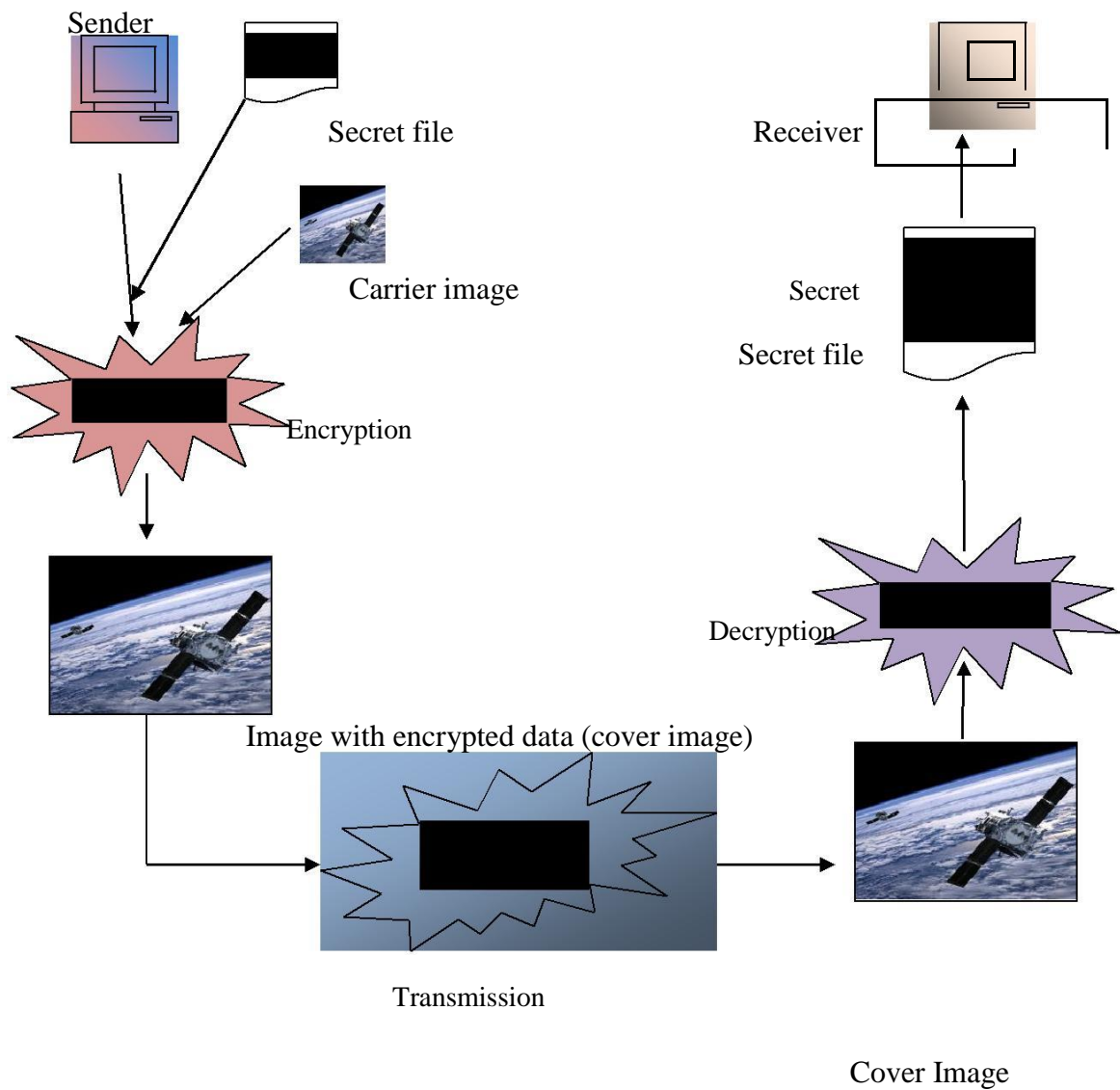


Figure 13: Block diagram for Steganography

The procedure for data hiding using steganographic application in this project is as follows

- The sender first uses the steganographic application for encrypting the secret message.
- For this encryption, the sender uses text document in which the data is written and the image as a carrier file in which the secret message or text document to be hidden.
- The sender sends the carrier file and text document to the encryption phase for data embedding, in which the text document is embedded into the image file. The procedure of encryption is discussed in the next phase.
- In encryption phase, the data is embedded into carrier file which was protected with the password
- Now the carrier file acts as an input for the decryption phase.
- The image in which data is hidden i.e. the carrier file is sent to the receiver using a transmission medium. E.g. Web or e-mail.
- The receiver receives the carrier file and places the image in the decryption phase.
- In the decryption phase, the original text document can be revealed using the appropriate password.
- The decryption phase decrypts the original text document using the least significant bit decoding and decrypts the original message.
- Before the encryption of the text, the message can be watermarked in order to avoid unauthorised modification.

As mentioned in the above block diagram, the data hiding and the data extracting will be done in three phases.

1. Encryption phase
2. Decryption phase
3. Transmission phase

3.1 ENCRYPTION PHASE:

The „Encryption phase“ uses two types of files for encryption purpose. One is the secret file which is to be transmitted securely, and the other is a carrier file such as image. In the encryption phase

the data is embedded into the image using „Least Significant Bit algorithm“ (LSB) by which the least significant bits of the secret document are arranged with the bits of carrier file such as image, Such that the

message bits will merge with the bits of carrier file. In this procedure LSB algorithm helps for securing the originality of image.

The encryption are divided into two types as discussed above

1. Symmetric encryption
2. Asymmetric encryption

The encryption pattern depends on the type of encryption we use. In this project, we am using the symmetric key encryption in which a single key is used. Symmetric encryption is shown using this block diagram.

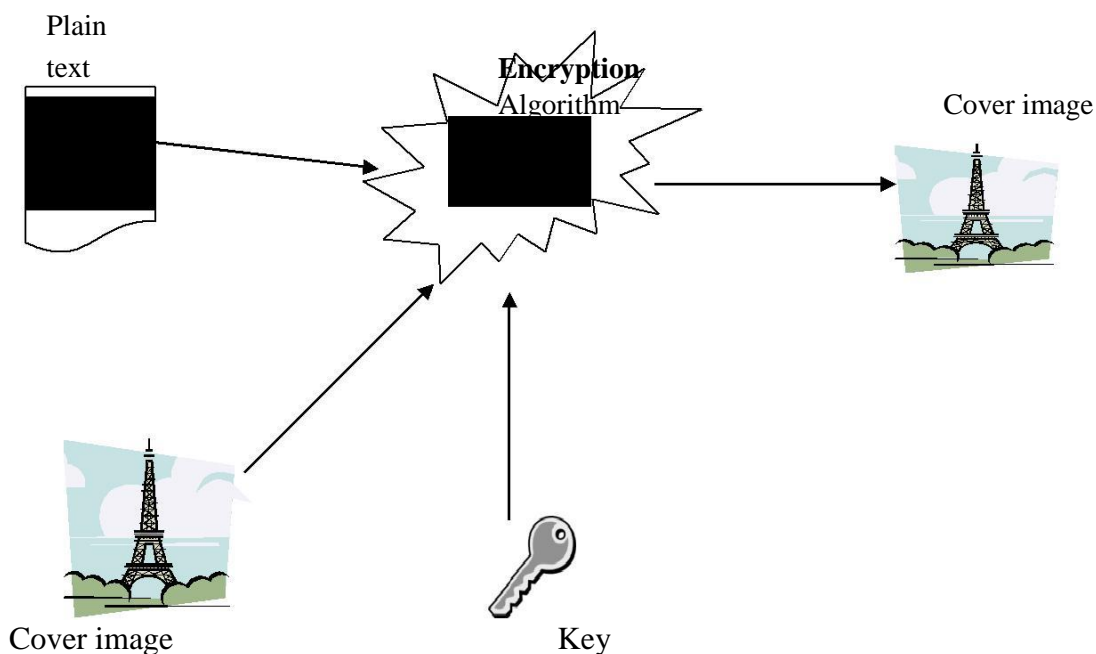


Figure 14: Encryption phase

3.2 TRANSMISSION PHASE:

The transmission phase is one of the important sections for sending the data to destination securely. The encryption section generates the cover image in which the data is embedded or hidden. This image is secured using the secret key. Usually we use e-mail or web for transferring the data. If the person hacks the e-mail or web and obtains the image, the secret key helps from unauthorized modification.

3.3 DECRYPTION PHASE:

The Decryption phase is reverse to encryption phase. In decryption phase, the carrier image in which the data is hided is given as an input file. The decryption phase uses the same password which was given for the encryption and decryption in order to secure from unauthorised access. After giving the correct password the decryption section uses the „Least Significant bit Algorithm“ (LSB) by which the encoded bits in the image is decoded and turns to its original state and gives the output as a text document as well as image.

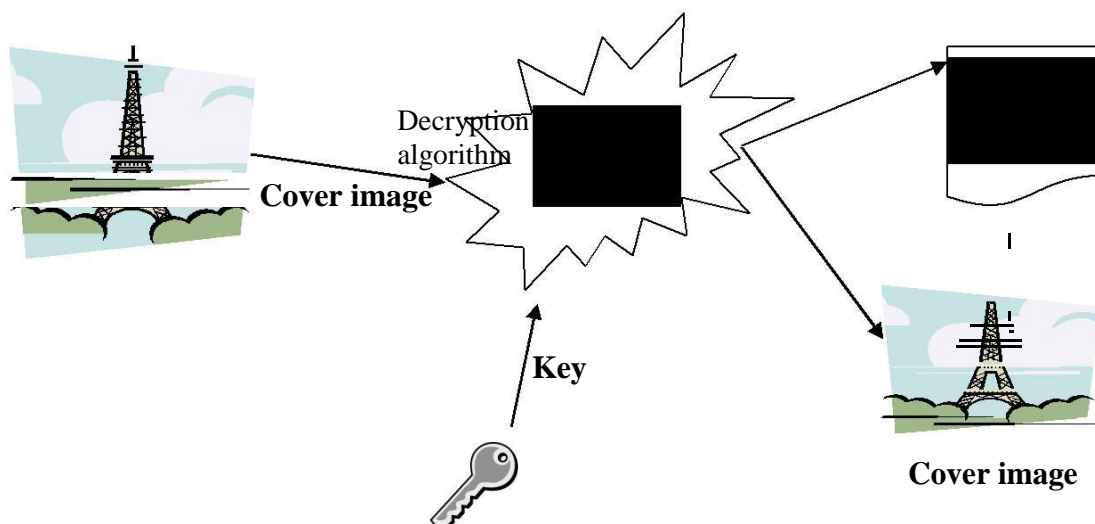


Figure 15: Decryption Phase

3.4 DATA FLOW DIAGRAMS:

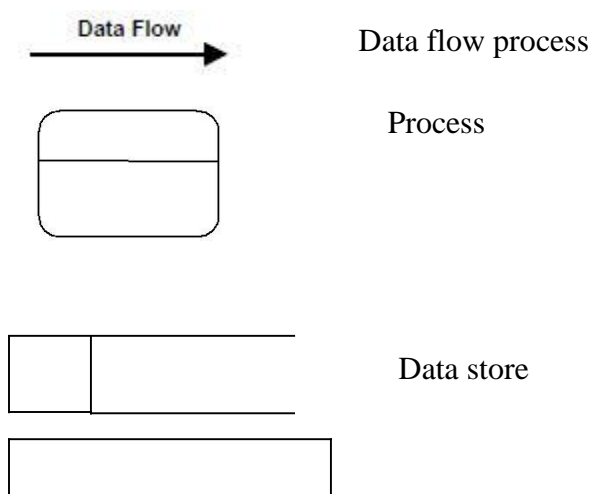
Data flow diagrams are the basic building blocks that define the flow of data in a system to the particular destination and difference in the flow when any transformation happens. It makes whole procedure like a good document and makes simpler and easy to understand for both programmers and non-programmers by dividing into the sub process.

The data flow diagrams are the simple blocks that reveal the relationship between various components of the system and provide high level overview, boundaries of particular system as well as provide detailed overview of system elements.

The data flow diagrams start from source and ends at the destination level i.e., it decomposes from high level to lower levels. The important things to remember about data flow diagrams are: it indicates the data flow for one way but not for loop structures and it doesn't indicate the time factors.

This section reveals about the data flow analysis which states about data that have been used, classification of data flow diagrams based on their functions and the other different levels used in the project.

The general notations for constructing a block diagram in this project are



Data flow processes:

It will define the direction i.e., the data flow from one entity to another entity.

Process:

Process defines the source from where the output is generated for the specified input. It states the actions performed on data such that they are transformed, stored or distributed.

Data store:

It is the place or physical location where the data is stored after extraction from the data source.

Source:

It is the starting point or destination point of the data, stating point from where the external entity acts as a cause to flow the data towards destination (**Wilson, 2004**).

3.4.1 Constructing Data Flow Diagram

The „data flow diagrams“ can be constructed by dividing the process into different levels like DFD 0, DFD 1, DFD 2, etc., for constructing the data flow diagram. For this process, these simple steps are to be followed.

- The data flow diagram can be constructed only when the process have one data flow in and one data flow out.
- The process should modify the incoming data and outgoing data.
- The data store should not be alone, should be connected with one process at least.
- The external entities of the process should be involved with one data flow.
- In data process the data flow should be from top to bottom and from left to right.
- In the data flow diagram, the data stores and their destinations are named with capital letters and the data flow and process should be small capitalizing the starting letter.

These rules should be followed for constructing the data flow diagrams.

3.4.2 Data Flow Diagram Level 0

„DFD level 0“ is the highest level view of the system, contains only one process which represents whole function of the system. It doesn't contain any data stores and the data is stored with in the process.

For constructing DFD level 0 diagram for the proposed approach we need two sources one is for „source“ and another is for „destination“ and a „process“.

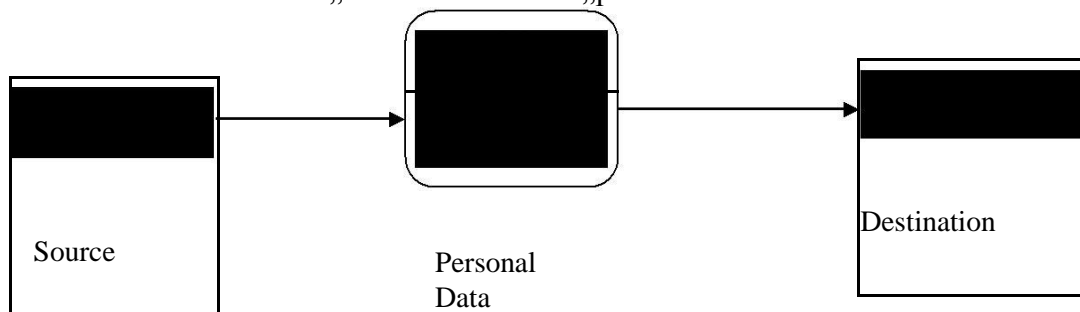


Figure 16: Data flow diagram level 0

DFD level 0 is the basic data flow process, the main objective is to transfer the data from sender to receiver after encryption.

3.4.3 Data Flow Diagram Level 1

For constructing „DFD level 1“, we need to identify and draw the process that make the level 0 process. In the project for transferring the personal data from source to destination, the personal data is first encrypted and processed and latter decrypted.

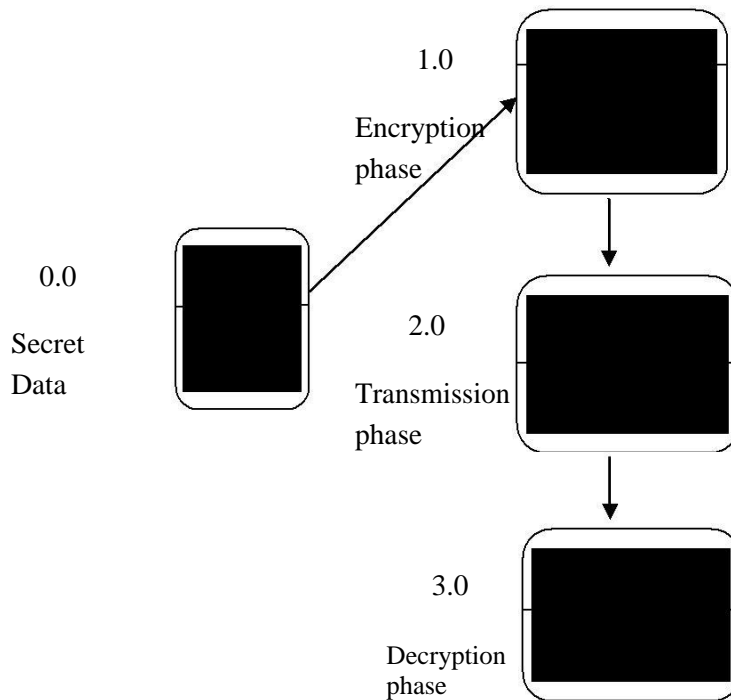


Figure 17: Data Flow Diagram level 1

In this data flow diagram, the secret data is sent to the encryption phase for embedding the data into the image for generating the carrier image. In the next phase the carrier image is sent to the decryption phase through the transmission phase. The final phase is the decryption phase where the data is extracted from the image and displays the original message

3.4.4 Data Flow Diagram Level 2

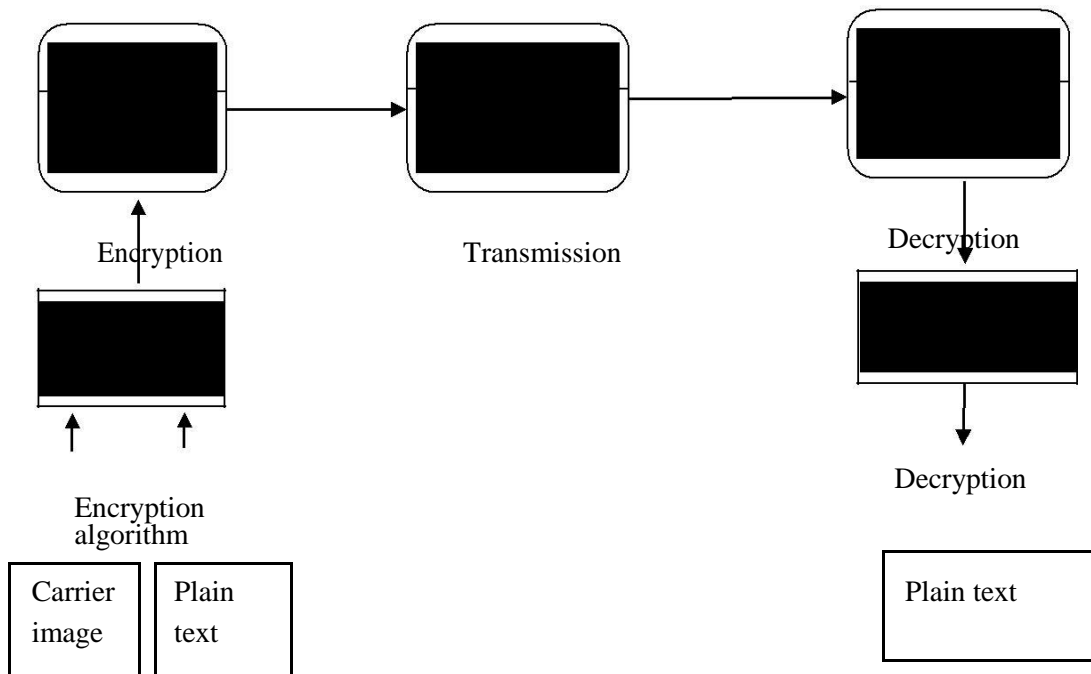


Figure 18: Data Flow Diagram level 2

The image and the text document are given to the encryption phase. The encryption algorithm is used for embedding the data into the image.

The resultant image acting as a carrier image is transmitted to the decryption phase using the transmission medium. For extracting the message from the carrier image, it is sent to the decryption section. The plain text is extracted from the carrier image using the decryption algorithm.

3.5 ACTIVITY DIAGRAM

The sender sends the message to the receiver using three phases. Since we are using the steganographic approach for transferring the message to the destination, the sender sends text as well as image file to the primary phase i.e., to encryption phase. The encryption phase uses the encryption algorithm by which the carrier image is generated. The encryption phase generates the carrier image as output.

activity diagram explains the overall procedure used for this project.

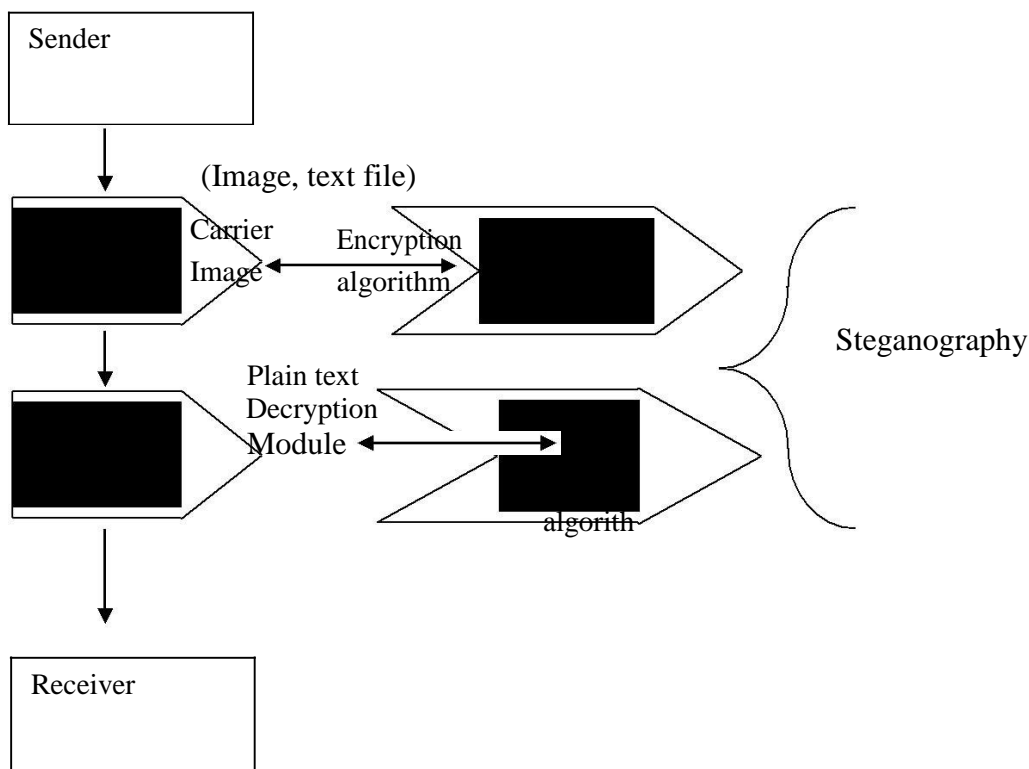


Figure 19: Activity diagram

The carrier image is given as input to the next phase i.e., to decryption phase. The decryption phase uses the decryption algorithm for decrypting the original text from the image so that the decryption phases generate plain text. The plain text is then sent to the receiver using the transmission media.

CHAPTER 4

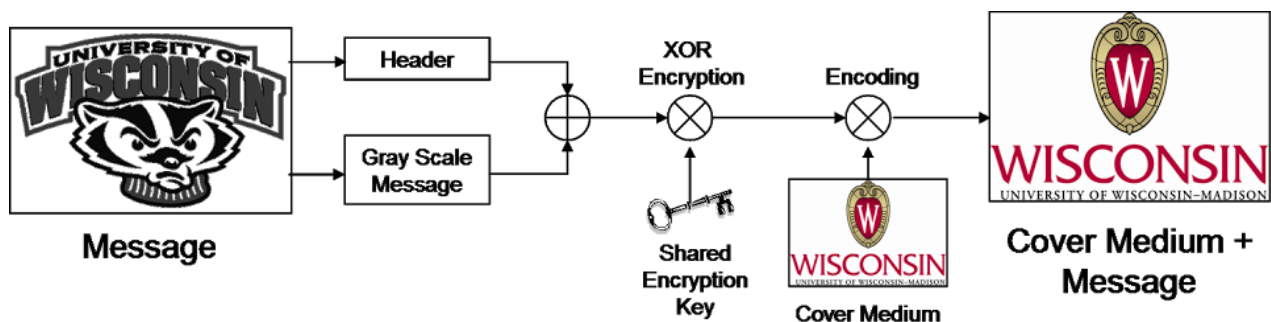
4.0 System Design

Steganography, which literally means "concealed writing" in Greek, is a centuries old science for passing hidden messages. Historically steganographic messages have been concealed using invisible inks or secret tattoos, but the digital revolution has provided a rich new area for concealing hidden messages within the underutilized bits in digital media (like image, video, and music files).

Steganography is also closely related to, and often uses, cryptography for improved message security. **Cryptography**, which literally means "hidden/secret writing" in Greek, uses a shared secret key, algorithm, or code to hide the message content from unauthorized access. Steganographic message security relies on its ability to not be detected while cryptographic message security relies on its ability to withstand decryption analysis.

4.1 Steganography process:

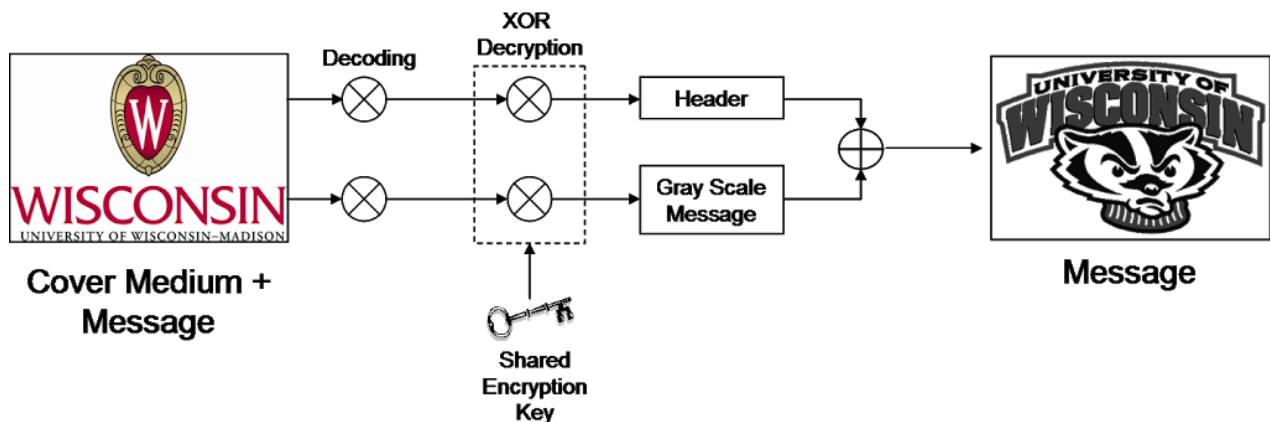
4.1.1 Encoding a Message:



1. Message Analyzed: Determine message type (image or text) and dimensions (Width/Height for images; Length for text).
 - o Color Images are converted to be Grayscale Images
2. Header Prepared: Header data added to beginning of message for Decoding Step.
 - o Image Header Format: 4 digits of Width Dimension followed by 4 digits of Height Dimension
 - o Text Header Format: 't' followed by 7 digits of Length Dimension

3. Encryption Applied: Header and Message encrypted using symmetric XOR encryption key.
4. Message Encoded: Encrypted Header/Message encoded onto Cover Medium.

4.1.2 Decoding a Message :



1. Message Decoded:
 - Header decoded/decrypted first to determine recovery cycles needed.
 - Algorithm recovers encrypted Message data.
2. Message Decrypted: Message decrypted using same symmetric XOR encryption key used earlier.
3. Message Recovered: Message put into viewable format.
 - Text Messages saved directly to .txt files.
 - Image Messages require dimensions from header to be put into .bmp files.

2. Sequential vs Pseudo-Random Encoding/Decoding

4.2 Sequential Encoding/Decoding:

Process:

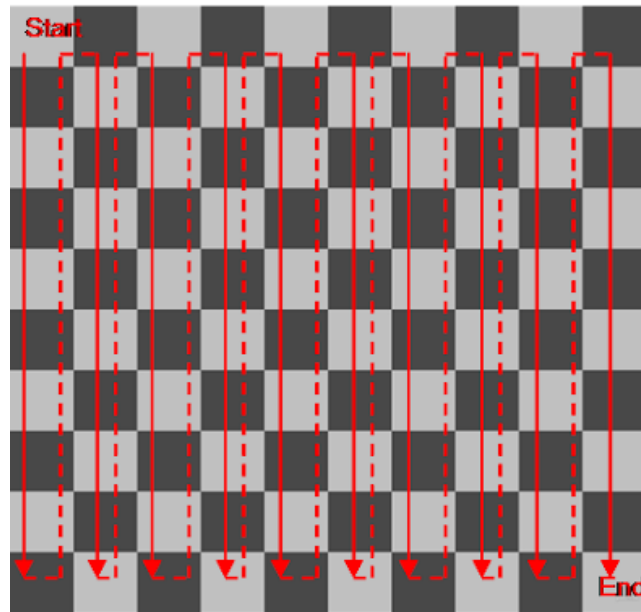
1. Message Data is Encoded/Decoded from some starting point (Typically upper left pixel)
2. Message Data is then Encoded/Decoded in a set unvarying pattern (Typically to adjacent pixels)

Advantages:

- Simple to Implement

Disadvantages:

- Easily Detected using Histogram Analysis
(Can also detect Message Size Directly)
- Slow and Tedious Recovery
(Usually implemented using counters; slows recovery process)
- Repeatable Encoding Method DECREASES Effectiveness of any Encryption Techniques



• Sequential Encoding Example

4.3 Pseudo-Random Encoding/Decoding:**Process:**

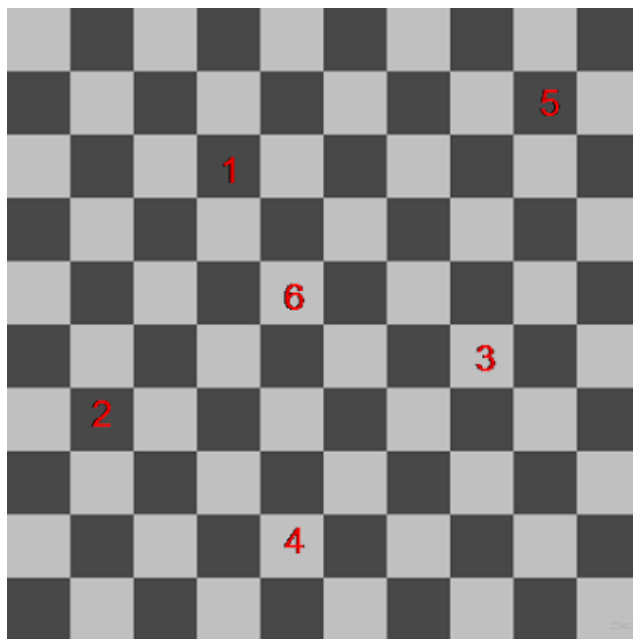
1. Pseudo-Random Number Generator Initialized
(typically no set starting point)
2. Message Data is then Encoded/Decoded based upon the pixel location determined by Random Number Generator
(typically no set pattern)

Advantages:

- No set Encoding/Decoding Pattern for Histogram Analysis to Detect
- Quicker Recovery Rate
(Usually implemented by Pre-Defining encoding pattern; more efficient for recovery process)
- Message size difficult to estimate

Disadvantages:

- Detectable using varying sized windows and localized Histogram Analysis



Pseudo-Random Encoding Example

4.4 Matlab Code and Examples

MATLAB Code Files

The MATLAB files we developed for this project are all available below in the 'Steganography_Code.zip' file. Below is a brief description of what these files do and how they work.

Steganography:

This function is an easy-to-use User Interface function that guides a user through the process of either encoding or decoding a message into or from within an image respectively.

Inputs / Outputs:

- No Inputs Required. User instead is prompted to provide necessary information.
- Automatically saves the cover image with encoded message as a Bitmap image or saves the decoded text or image message as a TXT or Bitmap file respectively.
- Returns encoded image or decoded message as a variable within MATLAB.

Features:

- User can select Encoding or Decoding.
- For Encoding, the user selects an image to hide the message within from a file list and then selects a text or image file message from a file list. The program then prompts the user to decide upon an encoding method, encryption key and random seed key before passing this information to the other functions. Finally the function will return the results as a variable in MATLAB as well as prompts user for an output image name and automatically saves the image to prevent message corruption.
- For Decoding, the user selects the image containing the hidden message from a file list. The program then prompts the user to provide the encoding method, encryption key, and random seed key before passing this information to the other functions for decoding. Finally the function will return the results as a variable in MATLAB as well as prompts the user for an output file name before it automatically saves the output message file.

Stegancoder:

This function determines the message type (text or image file), prepares header information to be used in the decoding stage, and sequentially encodes the message within the pixel values of the cover image.

Inputs / Outputs:

- Requires a cover image, text or image message, and encryption key as Inputs.
- Returns an image which has the message sequentially encoded as Output.

Features:

This function first determines the message type and length and encodes this information as header information (first 72 pixel values). Then the function sequentially encodes the message values across the Red, Green, and Blue Channels in a specific order defined within the function. This means that every message using this function is encoded from the Top Left pixel and is coded from Top to Bottom, Left to Right. This is considered less secure than a Random Encoding (see `Stegancoder_Rand`) but was simpler to code.

Stegandecoder:

This function recovers a sequentially encoded message that has been prepared using the `Stegancoder` file. This file takes in the cover image and encryption key, decodes the header to determine the message type and message length, and sequentially decodes and recovers the message from the pixel values of the cover image.

Inputs / Outputs:

- Requires the encoded cover image and encryption key as Inputs.
- Returns the decoded text or image message as an Output.

Features:

This function sequentially recovers the message values from the cover image by first isolating the header information (first 72 pixel values) to determine message type and length. The function then proceeds to decode the message using the length information from the header, uses the encryption key to decrypt the message and returns the message.

Stegancoder_Rand:

This function determines the message type (text or image file), prepares header information to be used in the decoding stage, and randomly encodes the message within the pixel values of the cover image.

Inputs / Outputs:

- Requires a cover image, text or image message, encryption key and random seed key as Inputs.
- Returns an image which has the message randomly encoded as Output.

Features:

This function first determines the message type and length and encodes this as header information (first 24 randomly encoded values). Then the function uses the `randperm` function to randomly select pixel locations to encode the message within. To do this the function determines the dimensions of the cover image, multiplies the dimensions together to provide the number of pixels available and uses `randperm` to randomly permute a list that includes values from 1 to the total pixel values available in a predictable and repeatable way by using the same random seed key value. This ensures that we don't overwrite message values in the cover image and can recover the message during the decoding stage (see `Stegandecoder_Rand`). The function then uses the `randperm` list to encode the message values in the cover image. This function is faster than the `Stegancoder` because the pixel locations are precomputed rather than encoded using counters as well as more secure because the message is encoded across the entire image instead of the left portion of the image.

Stegandecoder_Rand:

This function recovers a randomly encoded message that has been prepared using the Stegancoder_Rand file. This file takes in the cover image, encryption key and random seed key; decodes the header to determine the message type and message length; and decodes the randomly encoded message from the pixel values of the cover image.

Inputs / Outputs:

- Requires the encoded cover image, encryption key, and random seed key as Inputs.
- Returns the decoded text or image message as an Output.

Features:

This function uses the random seed key to initialize and recover the random pixel locations using the randperm function. The function first determines the cover image's dimensions to determine the amount of pixels available before determining the permuted pixel locations using randperm. Next the function recovers randomly encoded message values from the cover image by first isolating the header information (first 24 randomly encoded values) to determine the message type and length. The function then proceeds to decode the rest of the message using the length information from the header, uses the encryption key to decrypt the message and returns the message.

CHAPTER 5

5. Implementation

5.1 Code Analysis

```
%% STEP A: ENCODING VERSION
%% STEP 2A: Select "Canvas Image" and "Message File".
% First Get "Canvas" Image.
[FileName,PathName] = uigetfile({'*.jpg'; '*.png'; '*.gif'; '*.bmp'}, 'Select "Canvas Image"
to Hide Message. ');
img = imread( strcat(PathName,FileName) );

% Next get Message File
msg_type = input('Enter 1 for TEXT Message:\n');
if msg_type == 1
    [FileName,PathName] = uigetfile('*.txt', 'Select TEXT MESSAGE. ');
    testmsg = fopen( strcat(PathName,FileName) );
    [msg] = fscanf(testmsg, '%c');

else
    error('Invalid Message Type Selection');
end

%% STEP 3A: Prompt User for Encryption Key
enc_key = input('Please Enter an Encryption Key Between 0 - 255:\n');
if enc_key < 0 || enc_key > 255
    error('Invalid Key Selection');
end

enc_key = uint8(enc_key);

%% STEP 4A: Allow User to Select SEQUENTIAL or RANDOM Encoding Method
encode = input('Enter 1 for Sequential Encoding, 2 for Random Encoding:\n');
if encode == 1
    % SEQUENTIAL ENCODING: This only needs an Encryption Key Input.
    output = stegancoder(img,msg,enc_key);
elseif encode == 2
    % RANDOM ENCODING: This needs the Encryption Key AND Random Seed
    % Value.

    % Random Seed Value
    randSeed = input('Please Enter Random Seed Value Between 1 - 100:\n');
```

```

if randSeed < 1 || randSeed > 100
    error('Invalid Random Seed Value')
end

randSeed = uint8(randSeed);

% Final Output
output = stegancoder_Rand(img,msg,enc_key,randSeed);
else
    error('Invalid Encoding Selection');
end

%% STEP 5A: Write Canvas Image to .BMP File
% BMP, or bitmap format, was chosen because it DOES NOT use
% compression. JPEG compression destroys the message.
secfn = input('Enter File Name for Image + Message:\n','s');
nametest = ischar(secfn);
if nametest == 1
    imwrite(output, strcat(secfn, '.bmp'));
else
    error('Invalid File Name');
end

elseif enc_dec == 2
    %% STEP B: DECODING VERSION
    %% STEP 2B: Import "Canvas Image" With Hidden Message.
    [FileName,PathName] = uigetfile('*.bmp','Select "Canvas Image" With Hidden
Message. ');
    img = imread( strcat(PathName,FileName) );

    %% STEP 3B: Prompt User for Encryption Key
    enc_key = input('Please Enter an Encryption Key Between 0 - 255:\n');
    if enc_key < 0 || enc_key > 255
        error('Invalid Key Selection');
    end

    enc_key = uint8(enc_key);

    %% STEP 4B: Allow User to Select SEQUENTIAL or RANDOM Decoding Method
    decode = input('Enter 1 for Sequential Decoding, 2 for Random Decoding:\n');
    if decode == 1
        % SEQUENTIAL DECODING: This only needs an Encryption Key Input.
        output = stegandecoder(img,enc_key);
    end
end

```

```

elseif decode == 2
    % RANDOM DECODING: This needs the Encryption Key AND Random Seed
    % Value.

    % Random Seed Value
    randSeed = input('Please Enter Random Seed Value Between 1 - 100:\n');
    if randSeed < 1 || randSeed > 100
        error('Invalid Random Seed Value')
    end

    randSeed = uint8(randSeed);

    % Final Output
    output = stegandecoder_Rand(img,enc_key,randSeed);
else
    error('Invalid Encoding Selection');
end

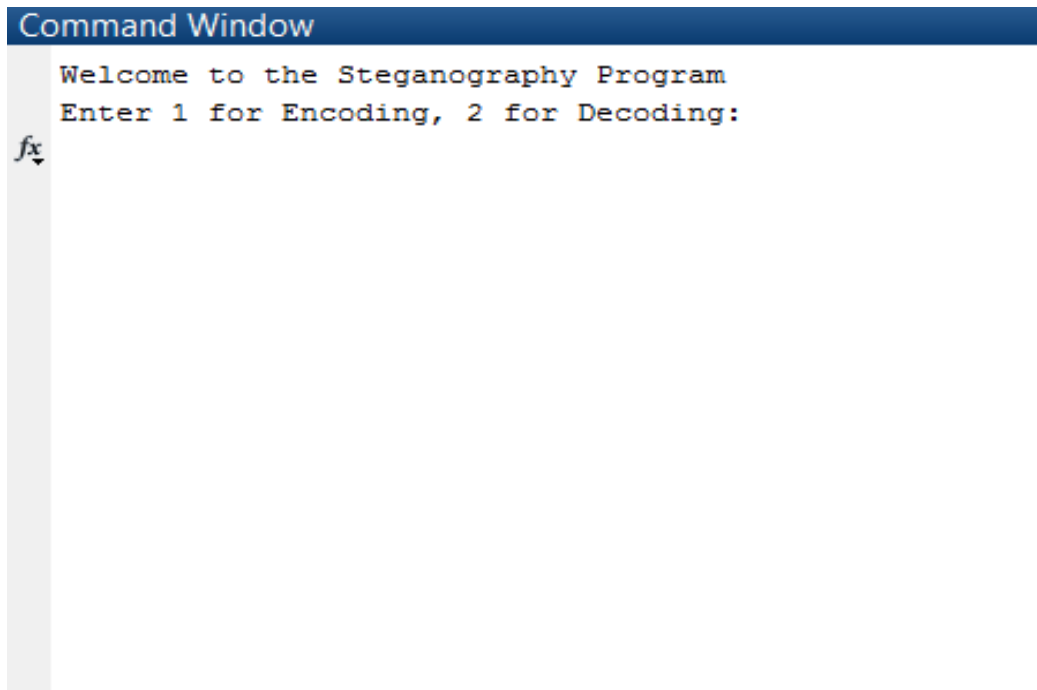
%% STEP 5B: Writing Message to .TXT or .JPG File
secfn = input('Enter File Name for Image + Message:\n','s');
nametest = ischar(secfn);
if nametest == 1
    msgtest = ischar(output);
    if msgtest == 1
        % TEXT Message CASE
        fid = fopen(strcat(secfn,'.txt'),'w');
        fwrite(fid,output,'char');
        fclose(fid);
    else
        % IMAGE Message CASE
        imwrite(output,strcat(secfn,'.bmp'));
    end
else
    error('Invalid File Name');
end

else
    error('Invalid Selection');
end

```

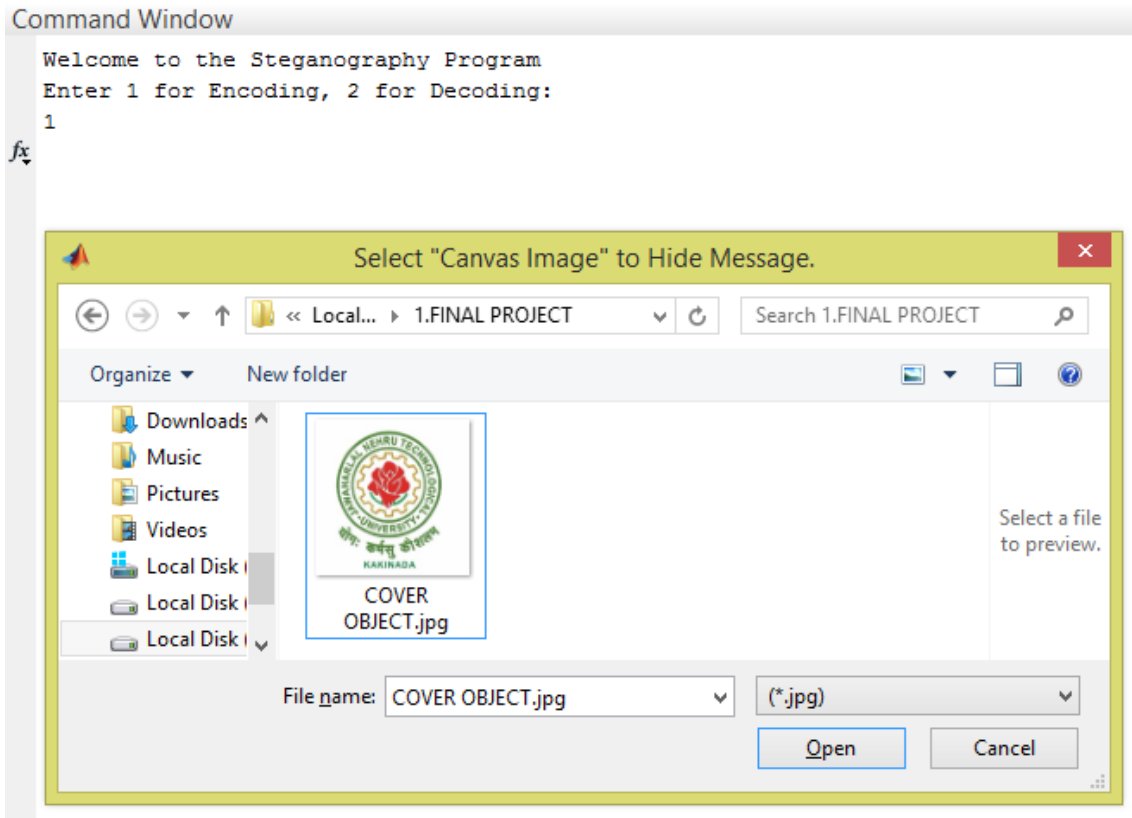
5.2 User Manual

This is the first screen which has two options – one is for encoding and another is for Decoding.



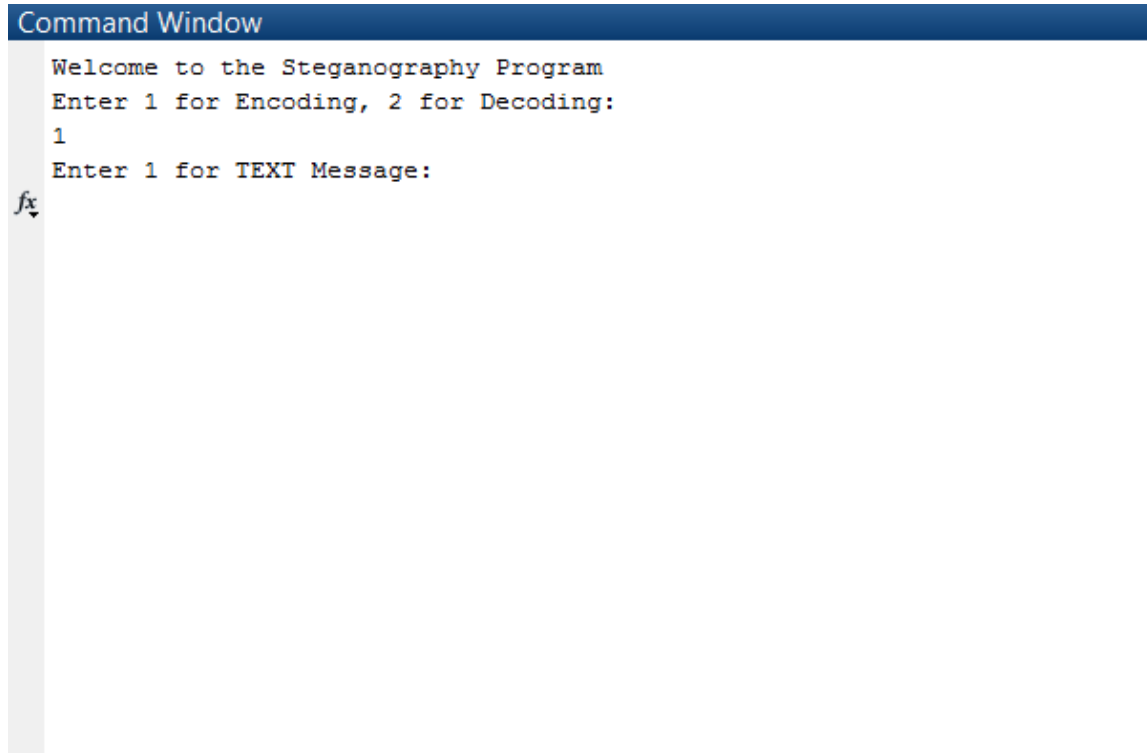
5.2.1 Encoding

1. For Encoding select “Canvas image” to hide message.



2. For load image click on button “Open”.

1. The image file will be loaded and is displays as follows. Next, type “1” to enter the TEXT message



The screenshot shows a command window titled "Command Window" with a dark blue header. The text inside the window is as follows:

```
Welcome to the Steganography Program
Enter 1 for Encoding, 2 for Decoding:
1
Enter 1 for TEXT Message:
```

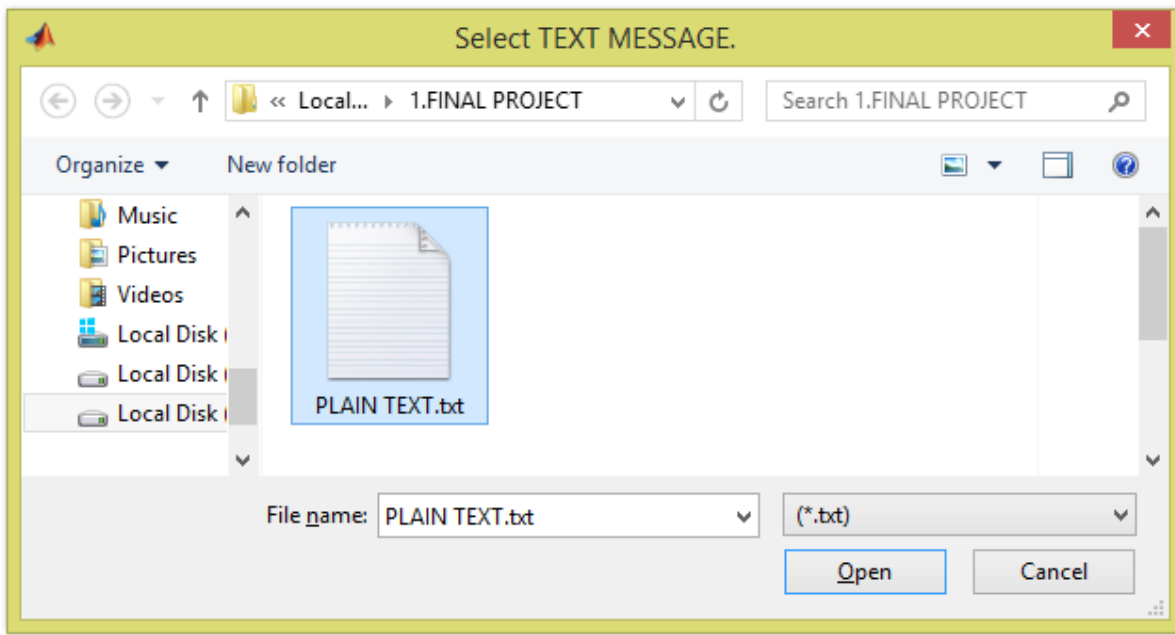
Below the text, there is a small icon of a cursor pointing to the right, indicating the current position in the command line.

Now, click on open to load TEXT file “PLAIN TEXT.txt”

Command Window

```
Welcome to the Steganography Program  
Enter 1 for Encoding, 2 for Decoding:  
1  
Enter 1 for TEXT Message:  
1
```

fx



The next step is to encrypt the file. Now, enter an encryption key between 0 and 255

```
Command Window
Welcome to the Steganography Program
Enter 1 for Encoding, 2 for Decoding:
1
Enter 1 for TEXT Message:
1
Please Enter an Encryption Key Between 0 - 255:
fx 222|
```

Now, enter “1” for Sequential encoding and “2” for Random encoding. If Sequential encoding is chosen, it will ask for file name as shown below

```
Command Window
Welcome to the Steganography Program
Enter 1 for Encoding, 2 for Decoding:
1
Enter 1 for TEXT Message:
1
Please Enter an Encryption Key Between 0 - 255:
222
Enter 1 for Sequential Encoding, 2 for Random Encoding:
1
Enter File Name for Image + Message:
fx STEGO OBJECT
```

Now, the process will be started as shown below

```
Command Window
254 254 254 254 254 254 254 254 254 254 254
254 254 254 254 254 254 254 254 254 254 254
254 254 254 254 254 254 254 254 254 254 254
254 254 254 254 254 254 254 254 254 254 254
254 254 254 254 254 254 254 254 254 254 254
254 254 254 254 254 254 254 254 254 254 254
254 254 254 254 254 254 254 254 254 254 254
254 254 254 254 254 254 254 254 254 254 254
254 254 254 254 254 254 254 254 254 254 254
254 254 254 254 254 254 254 254 254 254 254
254 254 254 254 254 254 254 254 254 254 254
254 254 254 254 254 254 254 254 254 254 254
254 254 254 254 254 254 254 254 254 254 254
254 254 254 254 254 254 254 254 254 254 254
254 254 254 254 254 254 254 254 254 254 254
254 254 254 254 254 254 254 254 254 254 254
254 254 254 254 254 254 254 254 254 254 254
254 254 254 254 254 254 254 254 254 254 254
```

If the Random encoding is chosen, it will ask for random seed value to choose between 0 and 100.

```
Command Window
Welcome to the Steganography Program
Enter 1 for Encoding, 2 for Decoding:
1
Enter 1 for TEXT Message:
1
Please Enter an Encryption Key Between 0 - 255:
222
Enter 1 for Sequential Encoding, 2 for Random Encoding:
2
Please Enter Random Seed Value Between 1 - 100:
fx 47
```

Now, After choosing the seed value, it asks for file name as shown below

Command Window

```
Welcome to the Steganography Program
Enter 1 for Encoding, 2 for Decoding:
1
Enter 1 for TEXT Message:
1
Please Enter an Encryption Key Between 0 - 255:
222
Enter 1 for Sequential Encoding, 2 for Random Encoding:
2
Please Enter Random Seed Value Between 1 - 100:
47
Enter File Name for Image + Message:
fx STEGO OBJECT
```

Now, the process will be started

Command Window												
244	231	149	101	101	81	79	82	78	83	81	82	81
246	244	161	102	102	78	79	75	79	80	81	81	79
248	247	196	95	98	85	77	74	82	80	82	80	77
248	242	227	98	87	85	73	76	78	79	83	77	74
253	244	232	125	85	81	76	75	76	79	82	77	73
252	247	234	182	96	92	82	84	83	76	81	76	75
252	248	246	216	127	97	86	84	84	79	82	77	75
252	250	241	239	169	93	90	80	85	82	83	80	78
252	250	240	246	209	111	95	84	85	77	84	84	81
252	251	248	254	241	172	95	80	80	80	85	86	84
252	252	255	250	248	226	119	80	84	86	88	88	85
252	252	255	246	242	240	181	109	92	87	89	89	87
252	252	252	255	250	245	232	139	93	94	90	88	86
254	255	255	255	251	248	242	214	115	103	94	90	83
254	255	255	255	255	253	248	236	153	116	96	91	82
254	254	255	254	255	254	250	245	186	119	100	92	83
254	254	255	253	253	254	252	241	206	126	105	96	86
254	254	253	253	252	255	254	241	223	143	111	100	91
254	254	251	254	253	254	251	244	234	157	116	105	94
254	254	252	254	253	250	246	243	235	162	121	109	97
254	254	254	250	249	252	248	247	241	170	123	111	99
255	254	250	247	253	253	255	254	247	181	127	115	102
255	255	252	250	254	255	254	250	251	195	133	118	96
255	255	250	251	253	255	250	249	249	198	122	111	91

The encoding process is completed.

5.2.2 Decoding

1. Select “2” for Decoding.

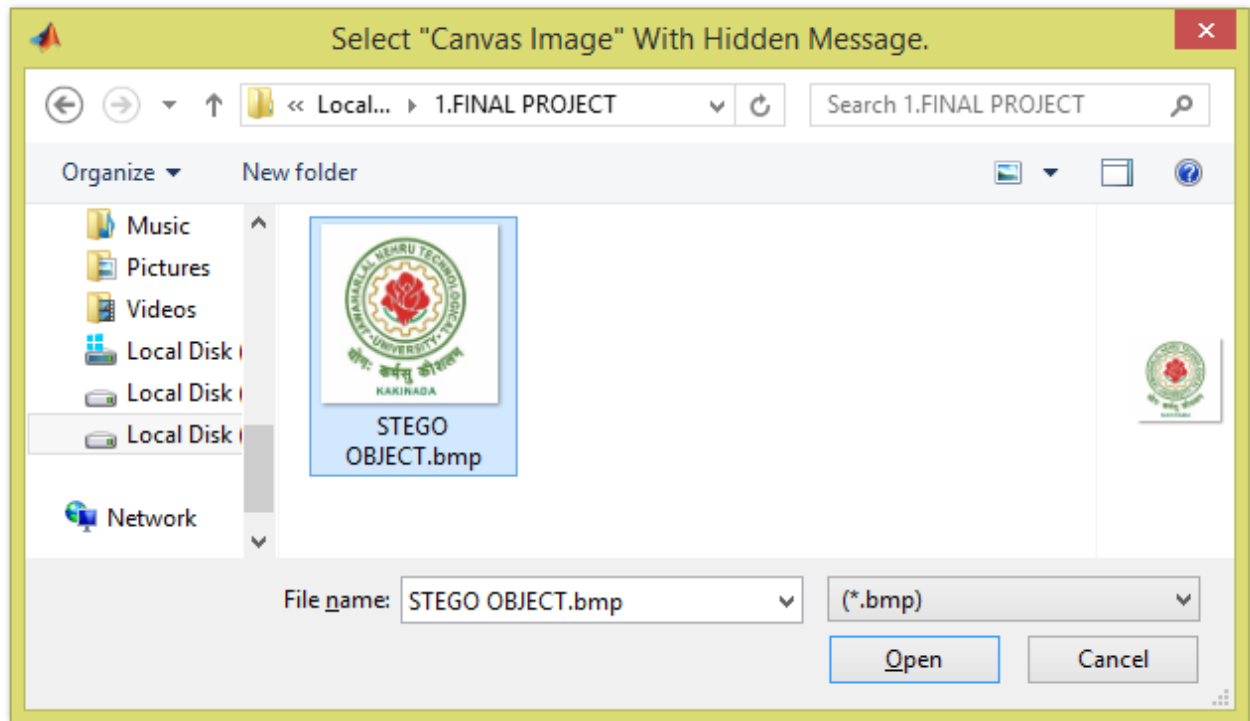
```
Command Window
Welcome to the Steganography Program
Enter 1 for Encoding, 2 for Decoding:
fx 2
```

Now, open the image file “STEGO OBJECT” as shown below

Command Window

```
Welcome to the Steganography Program  
Enter 1 for Encoding, 2 for Decoding:  
2
```

fx



After the loading of image, enter the encryption key which you have entered in the encoding process

Command Window

```
Welcome to the Steganography Program
Enter 1 for Encoding, 2 for Decoding:
2
Please Enter an Encryption Key Between 0 - 255:
fx 222
```

Now, enter “1” for sequential decoding and “2” for random decoding. If sequential decoding is chosen, it will ask to give some file name and then the output will be displayed as shown below

Command Window

```
Welcome to the Steganography Program
Enter 1 for Encoding, 2 for Decoding:
2
Please Enter an Encryption Key Between 0 - 255:
222
Enter 1 for Sequential Decoding, 2 for Random Decoding:
1
Enter File Name for Image + Message:
STEGO OUT

ans =

ECE PASS PERCENTAGE      : 92

CSE PASS PERCENTAGE      : 86

EEE PASS PERCENTAGE      : 81

CIVIL PASS PERCENTAGE    : 87
```

This is the output for the sequential decoding process. If random decoding is chosen it will ask for random seed value as shown below

Command Window

```
Welcome to the Steganography Program
Enter 1 for Encoding, 2 for Decoding:
2
Please Enter an Encryption Key Between 0 - 255:
222
Enter 1 for Sequential Decoding, 2 for Random Decoding:
2
Please Enter Random Seed Value Between 1 - 100:
fx 47
```

Now, enter the file name for the generated output and the output will be displayed as shown below

Command Window

```
Welcome to the Steganography Program
Enter 1 for Encoding, 2 for Decoding:
2
Please Enter an Encryption Key Between 0 - 255:
222
Enter 1 for Sequential Decoding, 2 for Random Decoding:
2
Please Enter Random Seed Value Between 1 - 100:
47
Enter File Name for Image + Message:
STEGO OUT

ans =

ECE PASS PERCENTAGE    : 92

CSE PASS PERCENTAGE    : 86

EEE PASS PERCENTAGE    : 81

CIVIL PASS PERCENTAGE  : 87
```

This is the output for random encoding process.

CHAPTER 6

6. RESULTS AND DISCUSSIONS

For designing the steganographic application, we worked on different phases like encryption, decryption and data transmission. An application for sending the personal data securely to the destination has been developed successfully.

The design phase is the primary phase, which gives a brief idea about the different levels used for developing an application with the help of block diagrams. The software is designed in a user friendly manner. So, it is simple to use for developing a prototype of the application.

The most important phase in the project is the execution phase. The execution phase is developed with the help of design phase. For executing the application, we worked on two sections: one is encryption and another is decryption. As we designed the program using Matlab platform, the next part is debugging the program. we faced some problems when writing the code, but at last we are successful in executing the program without errors. we used different approaches for testing the application, which helped me to know about the limitations.

In this project we mainly concentrated on embedding the data into an image. we have designed the steganographic application which embedded the data into the image. Normally, after embedding the data into the image, the image may lose its resolution. In the proposed approach, the image remains unchanged in its resolution as well in size. The speed of embedding the data into the image is also high in the proposed approach such that the image is protected and the data to the destination is sent securely. For the decryption phase, we have used the same Matlab programming language for the purpose of designing. we have used security keys like personal password for protecting the image from unauthorized modification, which improved the security level.

There are many steganographic algorithms available like JSteg, F5 and LSB algorithms. we have used the Least Significant Bit algorithm in designing the steganographic application because LSB algorithm works efficiently when we consider bit map images .bmp files. The speed of embedding is also high when

Using LSB compared to the JSteg algorithm. This approach is highly secured from stego attacks.

The final phase is transmitting the data to the destination. we have used keys section for protecting the data from unauthorized modification. The application uses the password as the reference such that whenever the image is sent using web sources like e-mails to the destination. For decryption, we need to use the same key which is used for encryption so that even if any unauthorized person hacks the web and access the image, it is not possible for decrypting the message which is embedded in it.

we have chosen image steganography because it is simple to use and its user friendly application. There are many applications for image hiding but the proposed approach is created using Matlab frame work which is easier for coding and the performance is better compared to other languages.

This project gave me good experience in dealing with the data security issues in theoretical as well as in technical domain and in Matlab programming as we used Matlab simulator software for designing steganographic application. we did the project in satisfactory level with the help and good guidance from my supervisor Smt. P. Pushpa latha

CHAPTER 7

7. CONCLUSION AND FUTURE WORK

In the present world, the data transfers using internet is rapidly growing because it is so easier as well as faster to transfer the data to destination. So, many individuals and business people use to transfer business documents, important information using internet. Security is an important issue while transferring the data using internet because any unauthorized individual can hack the data and make it useless or obtain information un-intended to him.

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 and is protected with a password which is highly secured.

The main intention of the project is to develop a steganographic application that provides good security. The proposed approach provides higher security and can protect the message from stego attacks. The image resolution doesn't change much and is negligible when we embed the message into the image and the image is protected with the personal password. So, it is not possible to damage the data by unauthorized personnel.

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.

The major limitation of the application is designed for bit map images (.bmp). It accepts only bit map images as a carrier file, and the compression depends on the document size as well as the carrier image size.

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 image formats like .bmp, .jpeg, .tif etc., in the future. The security using Least Significant Bit Algorithm is good but we can improve the level to a certain extent by varying the carriers as well as using different keys for encryption and decryption.

8.REFERENCES

Alfred J, M et al., 1996. *Hand book of applied Cryptography*. First edn.

Ali-al, H. Mohammad, A. 2010. Digital Audio Watermarking Based on the Discrete Wavelets Transform and Singular Value Decomposition, *European Journal Of Scientific Research*, vol 39(1), pp 231-239.

Amirthanjan,R. Akila,R & Deepikachowdavarapu, P., 2010. A Comparative Analysis of Image Steganography, *International Journal of Computer Application*, 2(3), pp.2-10.

Arnold, M. 2000. Audio watermarking: Features, applications and algorithms, *Proceeding of the IEEE International Conference on Multimedia and Expo*, pp 1013-1016.

Bandyopadhyay, S.K., 2010. An Alternative Approach of Steganography Using Reference Image. *International Journal of Advancements in Technology*, 1(1), pp.05-11.

Bloom,J. A. et al.,2008. *Digital watermarking and Steganography*. 2nd ed. Morgan Kaufmann.

Bishop, M., 2005. *Introduction to computer security*. 1st ed. Pearson publications.

Cachin, C., 2004. Information: Theoretic model for steganography. *Work shop on information hiding, USA*.

Chan, C.K. Cheng, L.M., 2004. *Hiding data in images by simple lsb substitution: pattern recognition*.vol 37. Pergamon.

Cox, I. Miller, M. Bloom, J. Fridrich, J & Kalker, T. 2008. *Digital watermarking and Steganography*. 2nd Ed. Elsevier.

Cummins, J. Diskin, P. Lau, S. & Parett, R., 2004. Steganography and digital watermarking. *School of computer science*. Vol 1.