Abstract

Steganography is an ability of concealing information inside the cover in such a way it looks like simple cover. The prime goal is to data hiding without much deteriorating the quality of the picture under consideration. Locating the best position in an image that will achieve the desired goal is a critical design issue. This approach focuses first on encrypting the secret data and then hiding the existence of the cipher so that any attacker would never know that a secret message is being passed over the channel. This hiding is done using steganographic technique using image as a cover media for reducing suspicions. In this method, secret data is initially encrypted using Advance Encryption Standard (AES) and steganography scheme hiding the cipher text into a grey cover image using Data Hiding Technique. The encryption key is sent securely through the exposed unsecure channel using Diffie-Hellman Key Exchange Protocol. On reception of the stego image, it undergoes extraction process. The extraction model is actually the reverse of the embedding model

Table of Contents

Abstract i
Contents..... ii
List Of Figures v
Abbreviations, Notations and Nomenclature vi
Chapter 1 Introduction 1
Chapter 2 Literature Survey 3
2.1 Steganography and Types 4
2.2 Various Steganography Techniques..... 4
2.2.1 Spatial Domain Technique 4
2.2.2 Transform Domain Technique 5
2.3 Encryption Methods 6
2.3.1 DES (Data Encryption Standard) 6
2.3.2 AES (Advanced Encryption Standard) 7
2.3.3 Triple DES..... 8
2.4 Key Exchange Algorithm..... 8
2.4.1 Diffie-Hellman: 8
Chapter 3 Problem Statement..... 9
Chapter 4 Requirement Analysis..... 10
4.1 Software Requirement Specification Document (SRS) 10
4.1.1 Introduction 10
4.1.1.1 Purpose 10
4.1.1.2 Scope 10
4.1.1.3 Intended Audience..... 10
4.1.1.4 Definitions, Acronyms, and Abbreviations...... 11
4.1.1.5 References 11
4.1.2 The Overall Description 11
4.1.2.1 Product Perspective 11
4.1.2.2 Product Functions..... 11
4.1.2.3 User Characteristics..... 12
4.1.2.4 Operating environment..... 12

4.1.2.5 Design and implementation constraints 12
4.1.3 External Interfaces Requirements: 12
4.1.3.1 User Interfaces........ 12
4.1.3.2 Hardware Interfaces 12
4.1.4 System Features........ 12
4.1.4.1 Enter data........ 12
4.1.4.2 Choose image 13
4.1.4.3 Encrypt data........ 13
4.1.4.4 Encode........ 14
4.1.4.5 Send........ 15
4.1.4.6 Decode........ 15
4.1.4.7 Decrypt 16
4.1.5 Other Non-Functional requirements........ 16
4.1.5.1 Performance Requirements 16
4.1.5.2 Safety:........ 16
4.1.5.3 Security: 17
4.1.5.4 Reliability:........ 17
4.1.5.5 Availability:........ 17
4.2 Use Case Diagram 18
4.3 System Activity Diagram 19
Chapter 5 Project Design........ 20
5.1 Sequence Diagram 20
5.2 Class Diagram 22
Chapter 6 Implementation Details........ 23
6.1 System Block Diagram........ 23
6.2 AES Encryption........ 24
6.3 Embedding........ 25
6.4 Peak Value Insertion 27
6.5 De-Embedding........ 27
6.6 AES Decryption 28
6.7 Diffie-Hellman Key Exchange Protocol 29
6.8 Snapshots........ 31
6.8.1 Selection of send or receive 31
6.8.2 After selection user needs to authenticate. 31

6.8.3 User specifies the secret message and cover image ... 32

6.8.4 User selects the receive option ... 33

Chapter 7 Testing ... 34

Chapter 8 Results and Analysis ... 36

Chapter 9 Conclusion and Future Scope ... 40

Appendix ... 42

Acknowledgements ... 44

List of Figures

Figure 4.1 Sender Side 18
Figure 4.2 Receiver Side 18
Figure 4.3 Activity Diagram 19
Figure 4.4 Sequence Diagram 21
Figure 4.5 Class Diagram 22
Figure 6.1 System Block Diagram 23
Figure 6.2 AES Encryption Model 24
Figure 6.3 Histogram of Barbara Image 25
Figure 6.4 Shift of Histogram 26
Figure 6.5 Embedding of data 26
Figure 6.6 Shifting of image example 27
Figure 6.7 Extraction of data 28
Figure 6.8 AES Decryption Process 29
Figure 6.9 Diffie- Hellman Key Exchange Protocol 30
Figure 6.10 Snapshot 1 31
Figure 6.11 Snapshot 2 31
Figure 6.12 Snapshot 3 32
Figure 6.13 Snapshot 4 33
Figure 8.1 Cover Image 36
Figure 8.2 Stego Image 37
Figure 8.3 Comparison of Orignal and Stego Image 38

List of Tables

Table 7.1 Test Cases 36

Abbreviations, Notations and Nomenclature

A

AES: Advanced Encryption Standard

ANSI: American National Standards Institute

APDT: Adjacent Pixel Difference Technique

D

DES: Data Encryption Standard

DCT: Discrete Cosine Transformation

F

FFT: Fast Fourier Transform

L

LSB: Least significant bit

P

PSNR: Peak signal-to-noise ratio

PP: Peak Point

S

SRS: Software Requirement Specification Document

T

TDES: Triple Data Encryption Standard

U

UML: Unified Modelling Language

Z

ZZ: Zero Point

Chapter 1
Introduction

Steganography is the art and science of writing hidden messages in such a way that no one apart from the intended recipient knows of the existence of the message; this is in contrast to cryptography, where the existence of the message itself is not disguised, but the content is obscured. The advantage of steganography over cryptography alone is that messages do not attract attention to themselves, to messengers, or to recipients.

Steganographic messages are often first encrypted by some traditional means, and then a cover text is modified in some way to contain the encrypted message, resulting in stego text. For example, the letter size, spacing, typeface, or other characteristics of a cover text can be manipulated to carry the hidden message; only the recipient (who must know the technique used) can recover the message and then decrypt it.

The project 'Steganography' provides means for secure data transmission and secure data storage network. Hereby, important files carrying confidential information can be stored in the server in an encrypted

Encryption is the process of encoding a message in such a way as to hide its contents. Modern Cryptography includes several secure algorithms for encrypting and decrypting messages. They are all based on the use of secrets called ***keys.*** A cryptographic key is a parameter used in an encryption algorithm in such a way that the encryption cannot be reversed without the knowledge of the key

Chapter 2
Literature Survey

Details explained by William Stallings [1] explained in detail the use of Advanced Encryption Standard and Diffie Hellman key exchanged protocol. Zhicheng Ni et.al in proposed a reversible data hiding algorithm for gray-scale images[2] . The results revealed better in terms of embedding capacity and security. The results were shown to have better embedding capacity and quality of image after hiding a data. Proposed novel reversible data hiding algorithm, which could recover the original image from marked image after extracting a hidden message. This method used the zero or minimum points of the histogram of an image for embedding a data into the original image and the results were found to be better than that of other algorithms in terms of PSNR. In [3], Y.-C. Li, C.-M. Yeh and C.-C. Chang *et al.* provided a brief overview of the similarity between neighbouring pixel technique with reversibility. In J Anita Christaline[4] shows more robustness and improved embedding capacity. In [5] Jagbir Singh *et al*. proposed the method of hiding a text message in grey scale image. The results were noticed to be better in terms of security as that of existing method by converting an original image into binary image.

2.1 Steganography and Types

Steganography is one of the most popular ways of sending vital information in a secret way. Steganography is the art and science of hiding information by embedding messages with in other seemingly harmless messages.

Various types of Steganography are as follows, based on the cover medium

- **Image Steganography**-Hiding Secret data into an image
- **Text Steganography**-Hiding Secret data inside Lexical texts
- **Audio Steganography**-Hiding Secret data inside the audio and video files

2.2 Various Steganography Techniques

There are many techniques for hiding information or messages in images in such a manner that the alterations made to the image are perceptually indiscreminable. The most common image steganographic approaches include :

2.2.1 Spatial Domain Technique

Spatial domain techniques embed messages in the intensity of the pixels directly. Thus the original pixels of the image are manipulated.There are many techniques for hiding information or messages in images in such a manner that the alterations made to the image are perceptually indiscreminable.

The most common image steganographic approaches include :

a) **Least Significant Bit-** Least significant bit (LSB) insertion is a common, simple approach to embedding information in a cover image. The least significant bit (in other words, the 8th bit) of some or all of the bytes inside an image is changed to a bit of the secret message.

b) **Peak and zero point Technique-**The image is analyzed by obtaining the histogram of the image and performing the manipulations on the histogram so as to embed the secret data.

2.2.2 Transform Domain Technique

Transform Domain methods hides messages in significant areas of cover image which makes them robust against various image processing operations like compression, enhancement etc. Many transform domain methods exist. The widely used transformation functions include

- Discrete Cosine Transformation (DCT)
- Fast Fourier Transform (DFT)
- Wavelet Transformation

The basic approach to hiding information with DCT[9], FFT[9] or Wavelet[9] is to transform the cover image, tweak the coefficients, and then invert the transformation. If the choice of coefficients is good and the size of the changes manageable, then the result is pretty close to the original.

1. **Discrete Cosine Transformation (DCT)-**DCT based data hiding used in the JPEG compression algorithm to transform successive 8x8- pixel blocks of the image from spatial domain to 64 DCT coefficients each in frequency domain. The least significant bits of the quantized DCT coefficients are used as redundant bits into which the hidden message is embedded.
2. **Discrete Fourier transforms (DFT)-**The Fourier transformed signal breaks down a signal into constituent sinusoids of different frequencies In other words: Transform the view of the signal from time-base to frequency-base.
3. **Wavelet Transformation**-The Haar Wavelet Transform [9]is the simplest of all wavelet transform. In this the low frequency wavelet coefficient are generated by averaging the two pixel values and high frequency coefficients are generated by taking half of the difference of the same two pixels. The four bands obtained are approximate band (LL), Vertical Band (LH), Horizontal band (HL), and diagonal detail band (HH).
4. **Spread Spectrum Domain Technique**

   In spread spectrum image steganography the message is embedded in noise and then combined with the cover image to produce the stego image. Since the power of the embedded signal is much lower than the power of the cover image, the embedded image is not perceptible to the human eye or by computer analysis without access to the original image.

2.3 Encryption Methods

Encryption is the most effective way to achieve data security. To read an encrypted file, you must have access to a secret key or password that enables you to decrypt it. Unencrypted data is called plain text , encrypted data is referred to as cipher text.

2.3.1 DES (Data Encryption Standard)

DES [1] is the block cipher which takes a fixed-length string of plaintext bits and transforms it through a series of complicated operations into another cipher text bit string of the same length. It is a symmetric encryption technique which means both sender and receiver use a shared key to encrypt and/or decrypt the data. The only problem with this technique is that if the key is known to others the entire conversation is compromised. In this, the block size is 64 bits it also uses a key to customize the transformation, so that decryption can supposedly only be performed by those who know the particular key used to encrypt. The key basically consists of 64 bits however, only 56-bits of these are actually used by the algorithm. Eight bits are used solely for checking parity, and are thereafter discarded. Hence the effective key length is 56-bits, and it is always quoted as such. Every 8th bit of the selected key is discarded i.e., positions 8,16, 24, 32, 40, 48, 56, 64 are removed from the 64-bit key leaving behind only the 56-bit key.

Advantages

1. DES has been around a long time (since 1977), even now no real weaknesses have been found: the most efficient attack is still brute force

2. DES is an official United States Government standard; the Government is required to re-certify, DES every five years and ask it be replaced if necessary.

DES has been re-certified in 1983, 1987, 1992.

3. DES is also an ANSI and ISO standard - anybody can learn the details and implement it.

4. Since DES was designed to run on 1977 hardware, it is fast in hardware and relatively fast in software.

Disadvantages

1. The 56-bit key size is the biggest defect of DES. Chips to perform one

million of DES encrypt or decrypt operations a second are available (in 1993).

A $1 million DES cracking machine can search the entire key space in about 7 hours.

2. Hardware implementations of DES are very fast; DES was not designed for software and hence runs relatively slowly.

3. As the technology is improving lot more day by day so there is a possibility to break the encrypted code, so AES is preferred than DES.

4. As we know in DES only one private key is used for encryption as well as for decryption because it is symmetric encryption technique so if we lost that key to decrypt the data then we cannot get the readable data at the receiving end.

2.3.2 AES (Advanced Encryption Standard)

AES [1] is a new cryptographic algorithm that can be used to protect electronic data. Specifically, AES is an iterative, symmetric-key block cipher that can use keys of 128, 192, and 256 bits, and encrypts and decrypts data in blocks of 128 bits (16 bytes). Unlike public-key ciphers, which use a pair of keys, symmetric-key ciphers use the same key to encrypt and decrypt data. Encrypted data returned by block ciphers have the same number of bits that the input data had.

Iterative ciphers use a loop structure that repeatedly performs permutations and substitutions of the input data.

Advantages:

1. The new AES will certainly become the de facto standard for encrypting all forms of electronic information, replacing DES.

2. AES-encrypted data is unbreakable in the sense that no known cryptanalysis attack can decrypt the AES cipher text without using a brute-force search through all possible 256-bit keys.

3. Security is no longer an afterthought in anyone's software design

and development process. AES is an important advance and using and

understanding it will greatly increase the reliability and safety of your software systems.

2.3.3 Triple DES

Triple DES (**3DES**) is the common name for the **Triple Data Encryption Algorithm** (**TDEA** or **Triple DEA**) symmetric-key block cipher, which applies the Data Encryption Standard (DES) cipher algorithm three times to each data block.

The original DES cipher's key size of 56 bits was generally sufficient when that algorithm was designed, but the availability of increasing computational power made brute-force attacks feasible. Triple DES provides a relatively simple method of increasing the key size of DES to protect against such attacks, without the need to design a completely new block cipher algorithm.

Disadvantage:

1. Processing times is more. Another disadvantage is when the file or hard drive crash, it's hard to recover.

2.4 Key Exchange Algorithm

Sending of secret message over a network is also important. To add more security ,there are different methods available key exchange algorithms are used for securely exchanging a key, which can be used for encryption between two or more parties such that the decryption can only be done by the desired recipient of the encrypted text.

2.4.1 Diffie-Hellman:

Diffie-Hellman key exchange protocol was developed to solve key distribution problem of symmetric key encryption. It is a public key cryptosystem. It generates secret key to be used for encrypting a message. It is a specific method of exchanging key.

Chapter 3
Problem Statement

To design a tool that performs secure transmission of secret data by encrypting it and hiding it in an image across an insecure network. The tool accepts the message from the user performs encryption using a key which is exchanged using Diffie-Hellman key exchange protocol and the resultant is embedded inside a cover image. On reception of this image the cipher text is extracted and the same key is used to decrypt the cipher and thus secret message is obtained.

Chapter 4
Requirement Analysis

4.1 Software Requirement Specification Document (SRS)

A Software Requirement Specification document(SRS) is a description of software system to be developed, laying out functional and non-functional requirements.

4.1.1 Introduction

The following subsections of the Software Requirements Specifications (SRS) document provide an overview of the entire SRS.

4.1.1.1 Purpose

To satisfy the requirements of reversible data hiding and to improve the hiding capacity.

4.1.1.2 Scope

Wide range of scope including military purposes, secure data transmission etc.

4.1.1.3 Intended Audience

Any person, organization that requires secure data transmission.

4.1.1.4 Definitions, Acronyms, and Abbreviations.

The definitions of the terms, which are used in this SRS document, are shown below

GUI	Graphical User Interfaces
SRS	Software Requirement Specification
AES	Advanced Encryption Standard

4.1.1.5 References

[1] IEEE STD 1233-1998, IEEE Guide for Developing System Requirements Specifications.

[2] IEEE STD 830-1998, IEEE Recommended Practice for Software Requirements Specifications.

4.1.2 The Overall Description

This section describes the general factors that affect the product and its requirements. It does not state specific requirements. Instead it provides a background for those requirements and makes them easier to understand.

4.1.2.1 Product Perspective

The Secure Data Hiding Using AES is an independent software. It is totally self-contained.

4.1.2.2 Product Functions

The use cases of Secure Data Hiding are explained:

- Enter secret data: The data which needs to be transmitted is entered.
- Select cover image: The image in which data has to be hid is chosen.
- Encrypt: The data is encrypted using AES.
- Choose Key: The key for encryption is chosen.
- Encode: The embedding of encrypted data into the cover image is carried out.
- Send: The image is sent
- Decode: The secret data is decoded from the cover image.
- Decrypt: The encrypted data is decrypted to get the actual data

4.1.2.3 User Characteristics

The person using this software does not need any kind of prior specific knowledge.

4.1.2.4 Operating environment

Microsoft Windows

4.1.2.5 Design and implementation constraints

The image in which data is to be hidden should be a gray scale image.

4.1.3 External Interfaces Requirements:

This section contains all the software requirements at a level of detail, that when combined with the system context diagram, use cases, and use case descriptions, is sufficient to enable designers to design a system to satisfy those requirements.

4.1.3.1 User Interfaces

The sender side application shall include option to enter the dat, to select a cover image, to encrypt the data, to send the data. The receiver side application will include option to decode the data, decrypt the data.

4.1.3.2 Hardware Interfaces

The system shall run on a Microsoft Windows based system.

4.1.4 System Features

4.1.4.1 Enter data

4.1.4.1.1 Brief Description

The secret data to be sent is entered.

4.1.4.1.2 Trigger

There are no triggers.

4.1.4.1.3 Precondition

The Application must be running and Input device like a keyboard should used.

4.1.4.1.4 Basic Flow

The data will be displayed in the text field

4.1.4.1.5 Alternate flow

There are no alternate flows.

4.1.4.1.6 Post Condition

The character count is reduced after each character been entered.

4.1.4.2 Choose image

4.1.4.2.1 Brief Description

A gray scale image is selected in which data is to be hidden.

4.1.4.2.2 Trigger

There are no triggers.

4.1.4.2.3 Precondition

The grey scale image must be used.

4.1.4.2.4 Basic Flow

Selected image will be displayed on the space provided in the user interface.

4.1.4.2.5 Alternate flow

There are no alternate flows.

4.1.4.2.6 Post Condition

The path of the chosen image will be displayed.

4.1.4.3 Encrypt data

4.1.4.3.1 Brief Description

The data is encrypted using AES.

4.1.4.3.2 Trigger

There are no triggers.

4.1.4.3.3 Precondition

The secret data must be provided by the user.

4.1.4.3.4 Basic Flow

The data entered by the user will be encrypted.

4.1.4.3.5 Alternate flow

There are no alternate flows.

4.1.4.3.6 Post Condition

A message will be displayed that the data has been encrypted.

4.1.4.4 Encode

4.1.4.4.1 Brief Description

The encrypted data is embedded inside the cover image.

4.1.4.4.2 Trigger

There are no triggers.

4.1.4.4.3 Precondition

The user data must be encrypted and the cover image of the desired format must be used.

4.1.4.4.4 Basic flow

The encrypted data will be embedded inside an image.

4.1.4.4.5 Alternate flow

There are no alternate flows.

4.1.4.4.6 Post Condition

A message will be displayed that the data has been embedded.

4.1.4.5 Send

4.1.4.5.1 Brief Description

The encoded image is securely send to the receiver side.

4.1.4.5.2 Trigger

There are no triggers.

4.1.4.5.3 Precondition

The secret data must be successfully embedded inside the cover image.

4.1.4.5.4 Basic Flow

The stego image will be send over the channel to the receiver.

4.1.4.5.5 Alternate flow

There are no alternate flows.

4.1.4.5.6 Post Condition

A message will be displayed at the sender side that the image has been sent.

4.1.4.6 Decode

4.1.4.6.1 Brief Description

The image received is decoded to obtain the encrypted data i.e the reverse process of encoding.

4.1.4.6.2 Trigger

There are no triggers.

4.1.4.6.3 Precondition

The stego image should be received successfully. Check whether the encrypted data obtained using decoding is same as the cipher text obtained during encryption.

4.1.4.6.4 Basic Flow

The received image will be decoded to obtain the embedded secret data.

4.1.4.6.5 Alternate flow

There are no alternate flows.

4.1.4.6.6 Post Condition

The obtained secret data will be decrypted

4.1.4.7 Decrypt

4.1.4.7.1 Brief Description

The encrypted data is decrypted to get the actual data.

4.1.4.7.2 Trigger

There are no triggers.

4.1.4.7.3 Precondition

The receipted image must be successfully decoded and the secret data should be retrieved.

4.1.4.7.4 Basic Flow

The obtained secret data will be decrypted

4.1.4.7.5 Alternate flow

There are no alternate flows.

4.1.4.7.6 Post Condition

The secret data will be displayed to the receiver.

4.1.5 Other Non-Functional requirements

4.1.5.1 Performance Requirements

Performance requirements define acceptable response times for system functionality.

- The load time for user interface screens shall take less time.
- Embedding process will be fast.

4.1.5.2 Safety:

There is no replication of secret data in the system

4.1.5.3 Security:

Since there is no replication and every time a new key is generated for transmitting.

4.1.5.4 Reliability:

Specify the factors that are required to establish required reliability of the software system at time of delivery.

4.1.5.5 Availability:

The system shall be available at all times.

4.2 Use Case Diagram

Use case diagram explains the different functionalities that can be performed by the user on both the sender side as well as the receiver side.

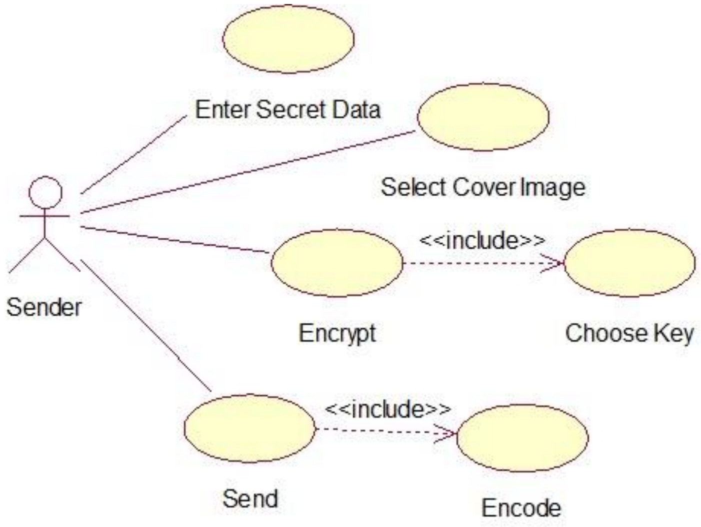


Figure 4.1 Sender Side

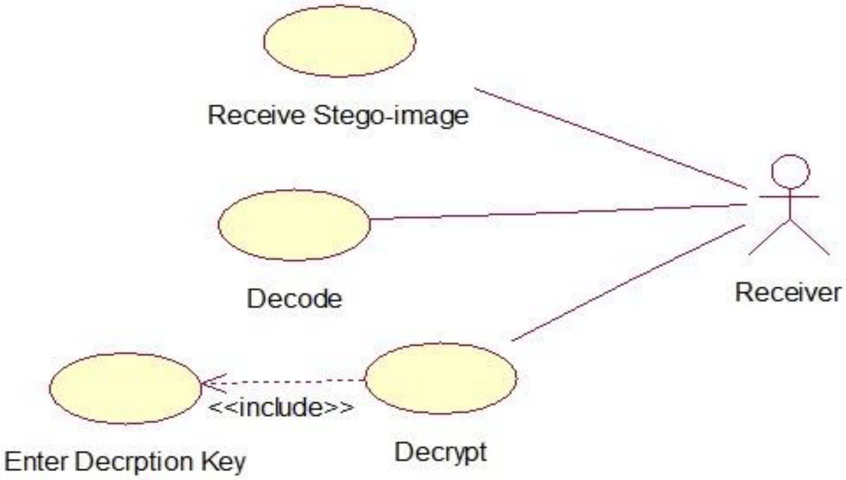


Figure 4.2 Receiver Side

4.3 System Activity Diagram

Activity diagram depicts various functions performed by the sender, the system and the receiver.

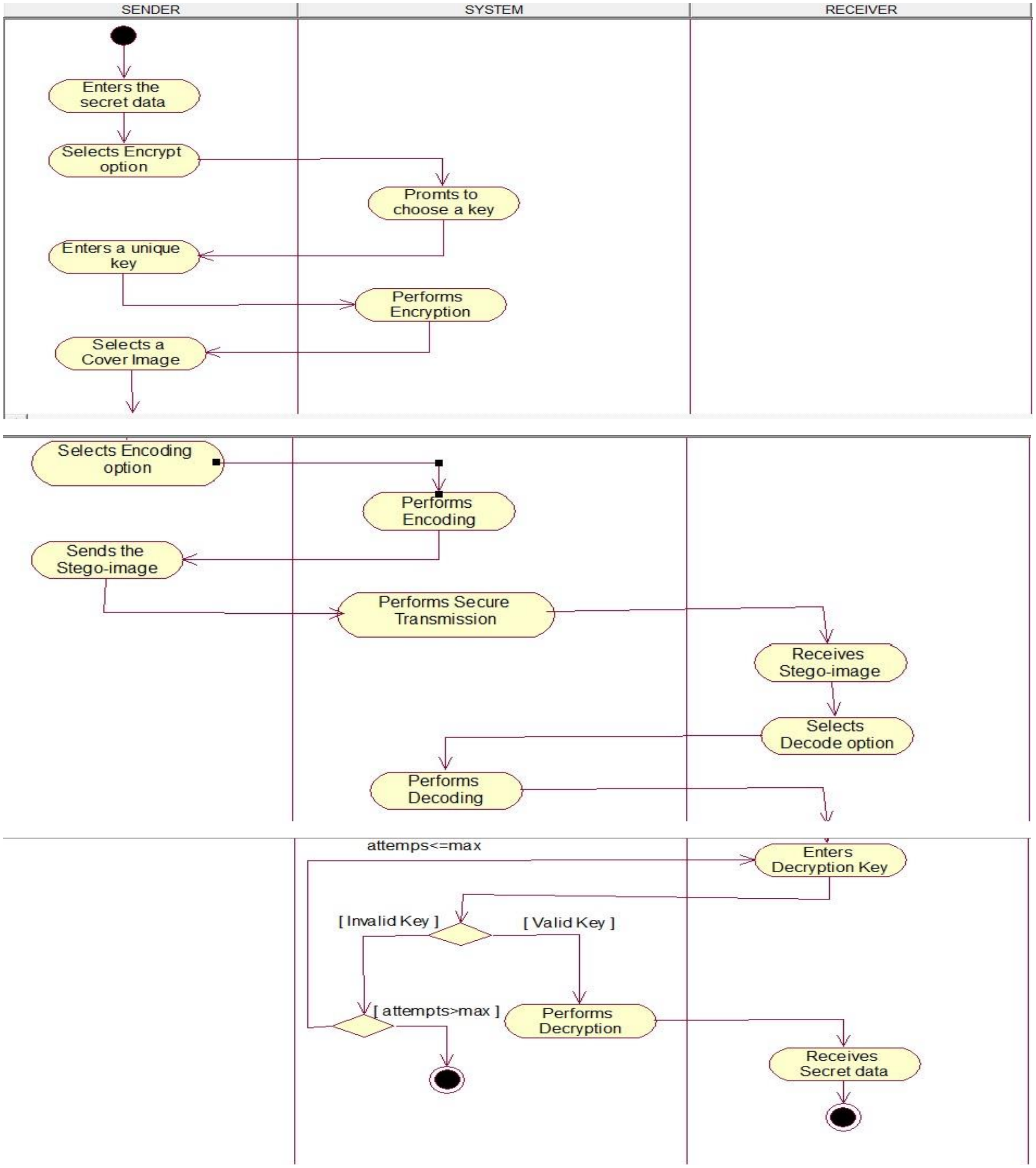


Figure 4.3 Activity Diagram

Chapter 5
Project Design

5.1 Sequence Diagram

Sequence diagram explains about the sequence in which a particular function is performed in the system. Following figure depicts the sequence of embedding the message into a cover image and delivering it successfully to the receivers side.

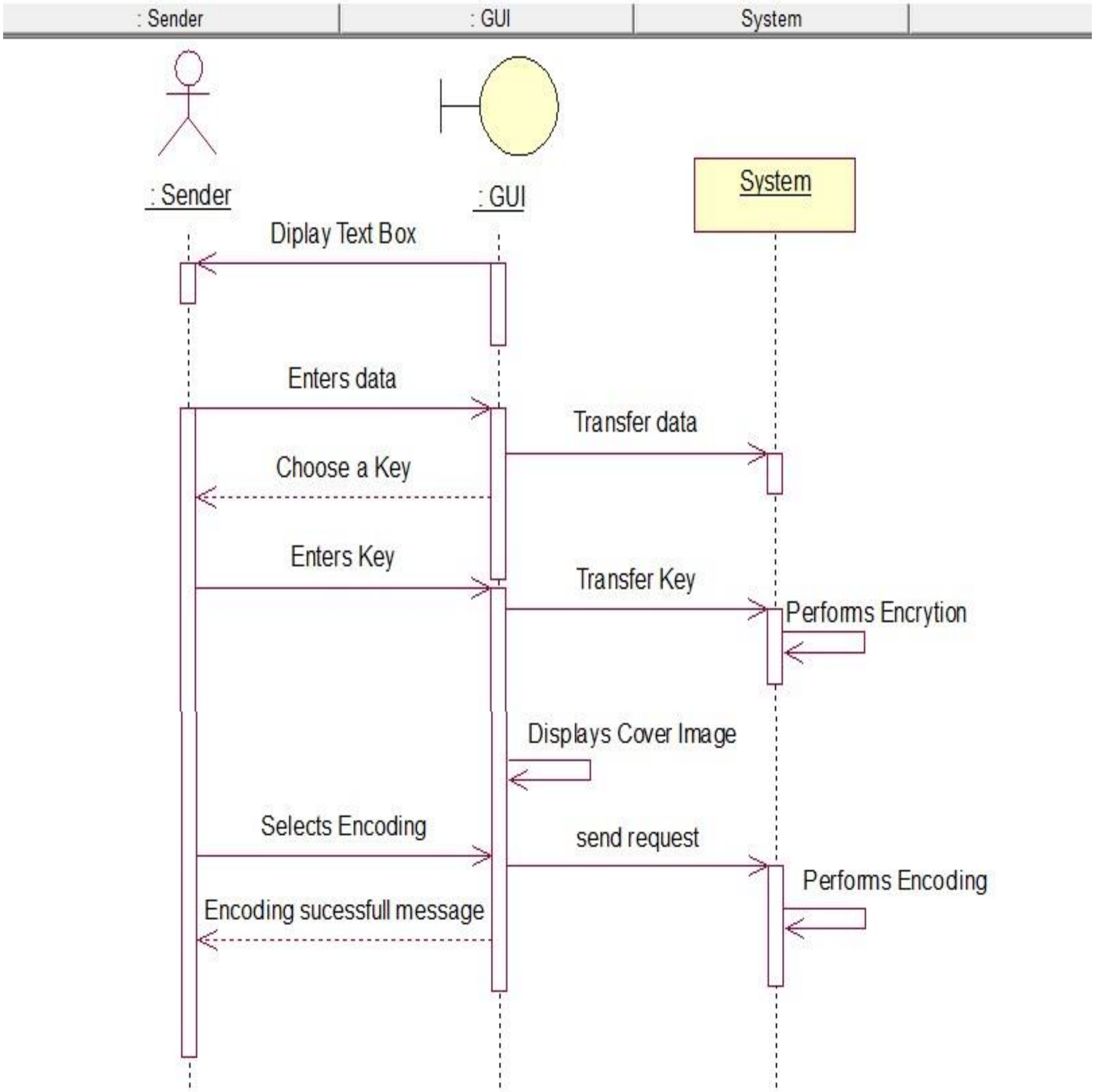


Figure 4.4 Sequence Diagram

5.2 Class Diagram

A class diagram is an illustration of the relationships and source code dependencies among classes in the UML.

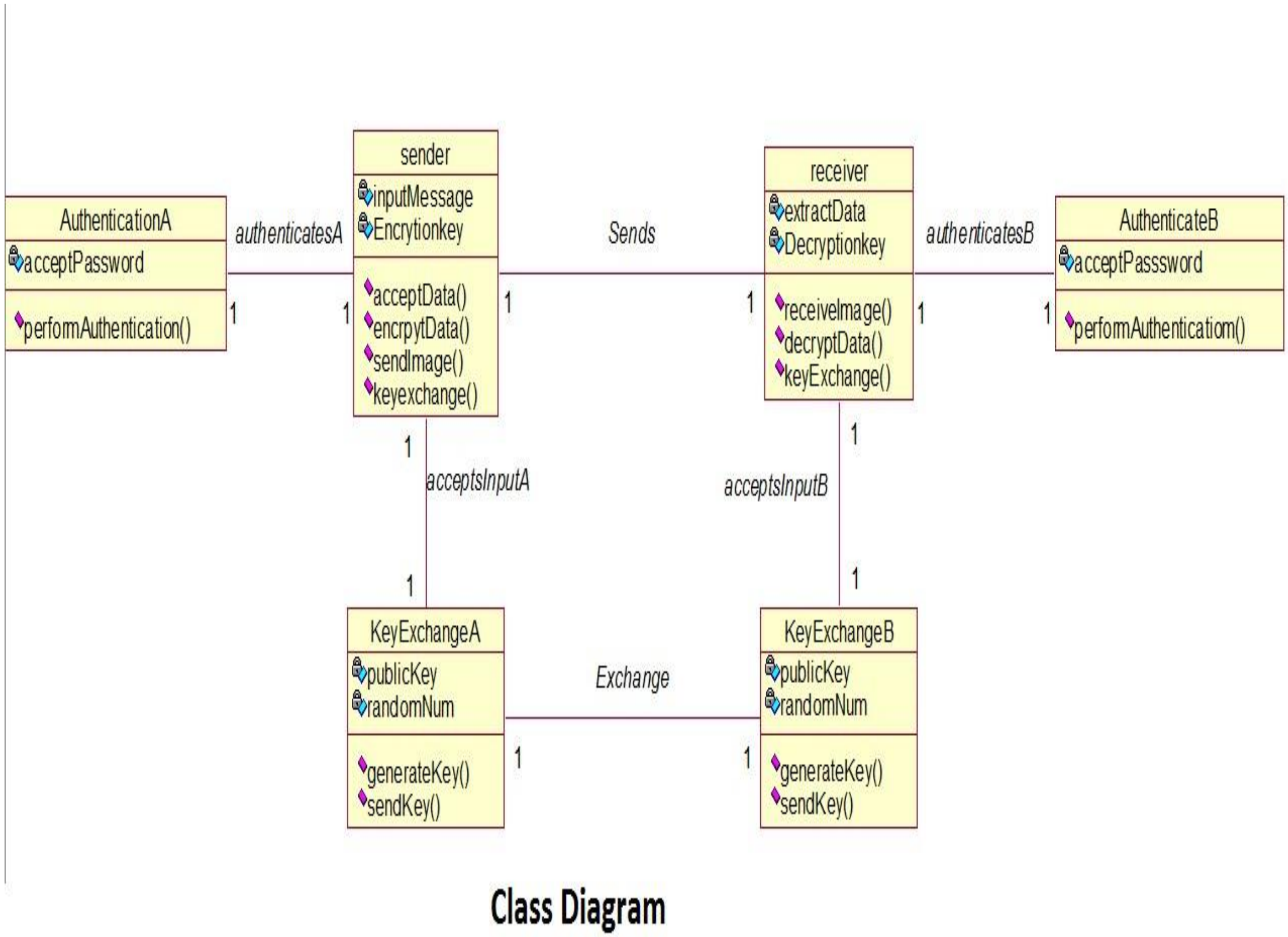


Figure 4.5 Class Diagram

Chapter 6
Implementation Details

6.1 System Block Diagram

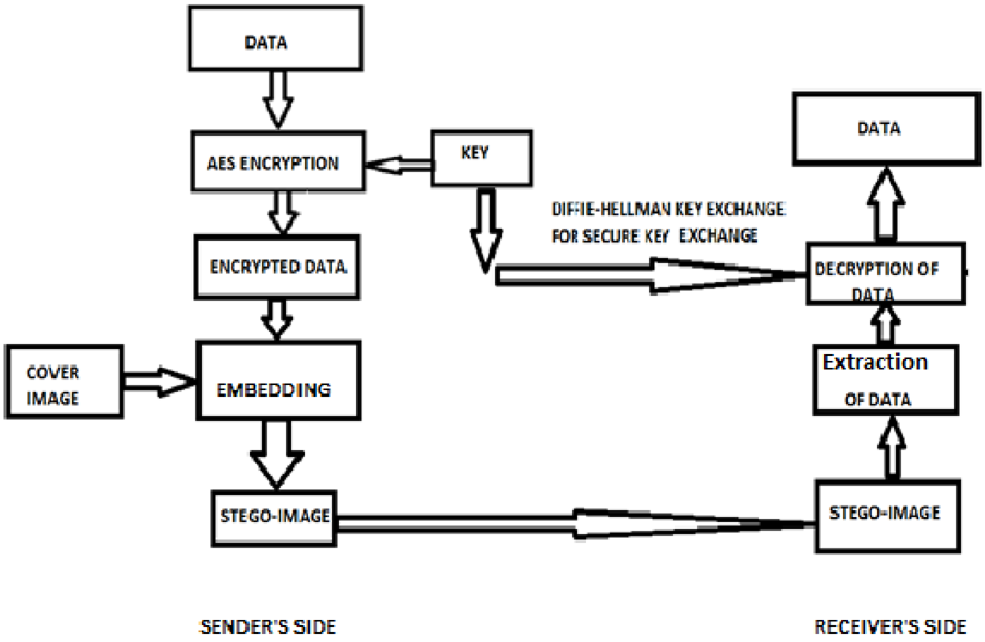
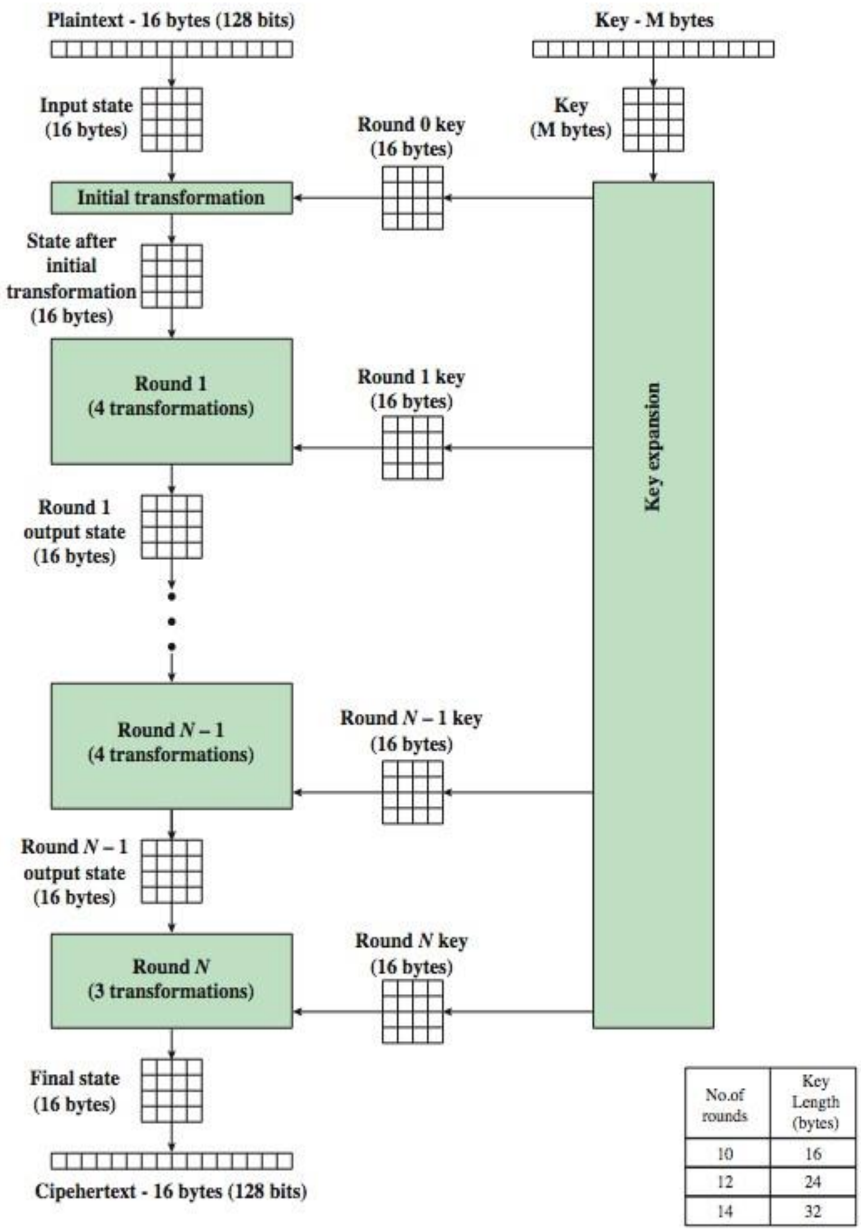


Figure 6.1 System Block Diagram

6.2 AES Encryption

Advanced Encryption Standard is a symmetric key encryption algorithm. It uses a block size of 128 bits and key lengths of 128, 192, 256 bits which depend on the number of rounds.AES uses cycle called as Rounds..The complete AES algorithm is explained using the following image:


Plaintext - 16 bytes (128 bits)
Key - M bytes
Input state (16 bytes)
Key (M bytes)
Round 0 key (16 bytes)
Initial transformation
State after initial transformation (16 bytes)
Round 1 (4 transformations)
Round 1 key (16 bytes)
Key expansion
Round 1 output state (16 bytes)
Round N – 1 (4 transformations)
Round N – 1 key (16 bytes)
Round N – 1 output state (16 bytes)
Round N (3 transformations)
Round N key (16 bytes)
Final state (16 bytes)
Cipehertext - 16 bytes (128 bits)

No.of rounds	Key Length (bytes)
10	16
12	24
14	32

Figure 6.2 AES Encryption Model [1]

6.3 Embedding

The cipher text obtained from AES is embedded inside the cover image.
The algorithm is as follows:

- **Step 1**: For a given image, Fig. 1 shows the histogram of the Barbara image.

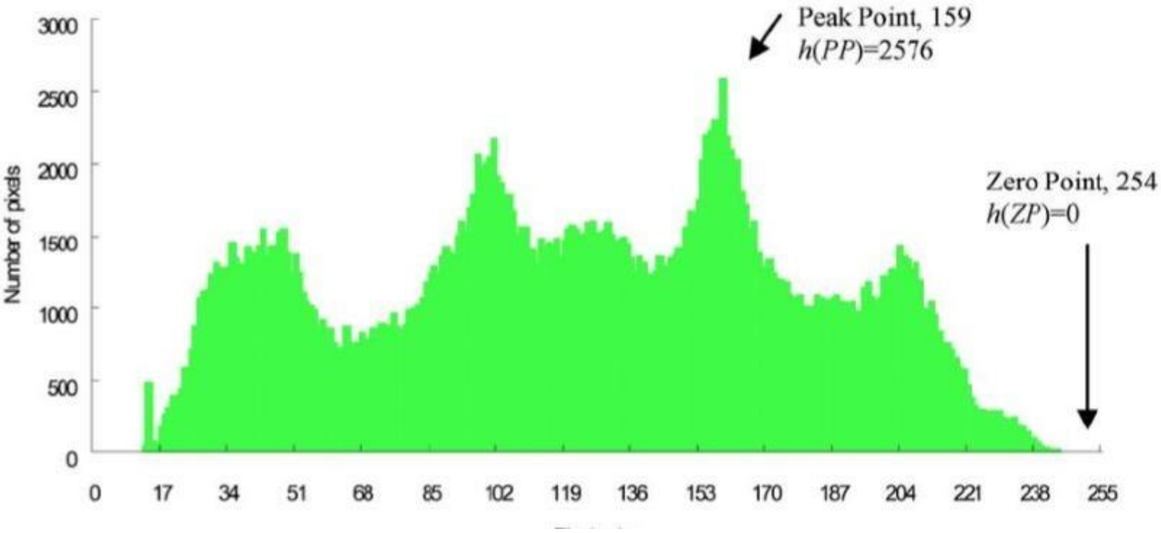


Figure 6.3 Histogram of Barbara Image [2]

- **Step 2**: Find and store the most frequent and least frequent pixel values.

1. For example, in Fig. 1, the pixel value 159 occurs 2576 times, denoted as h(159) = 2576, which is the maximum number of occurrences, and is called the peak point
2. The pixel value 254 does not appear in the Barbara image, which includes the minimum pixel number, 0, called the minimum point or zero point.
3. If there is no zero point, make a minimum point by cleaning the data and store the pixel information.

- **Step 3**: Scan the cover image once in a sequential order.

1. If PP > ZP, then shift each pixel value in the range, [ZP+1, PP−1], to the left-hand side by decreasing the pixel value by one unit.
2. If PP < ZP, then shift each pixel value in the range, [PP + 1, ZP − 1], to the right-hand side by increasing the pixel value by one unit.

For example, in Fig. 1, PP < ZP since 159 < 254, each pixel value in the range, [160, 253], is increased by one.

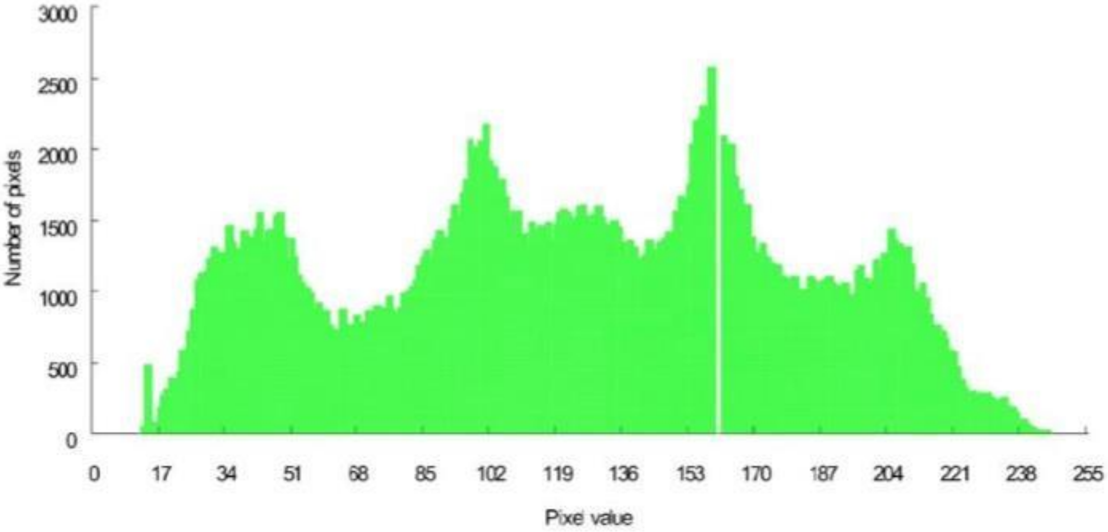


Figure 6.4 Shift of Histogram [2]

- **Step 4**: Scan the whole image once again in the same sequential order to embed data. After scanning the pixel with the peak point value, embed a bit of the hidden data. if the embedded bit is "1", then shift the pixel value from PP to ZP by one; otherwise, the pixel value does not change.

Given the Barbara image, if it selects just one pair of peak point and zero point, it can at most embed 2576 bits.

It can select multiple pairs of peak points and zero points to increase its embedding capacity.

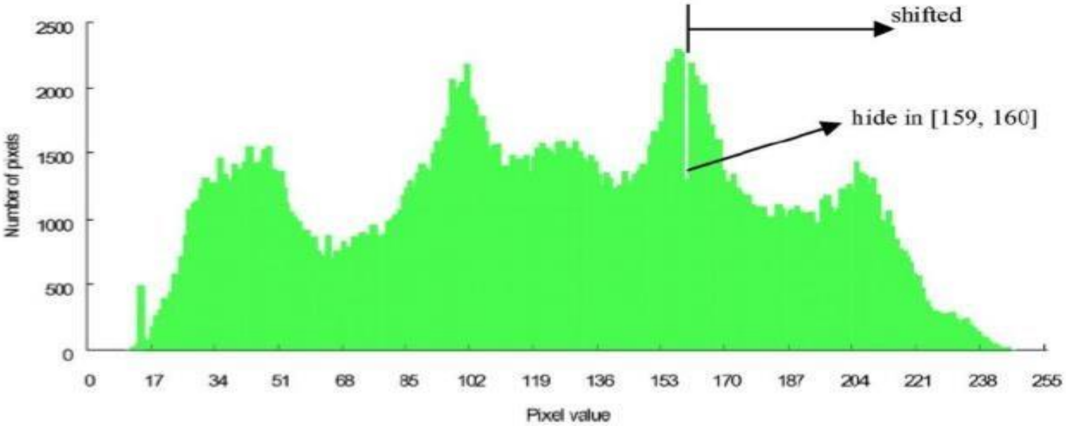
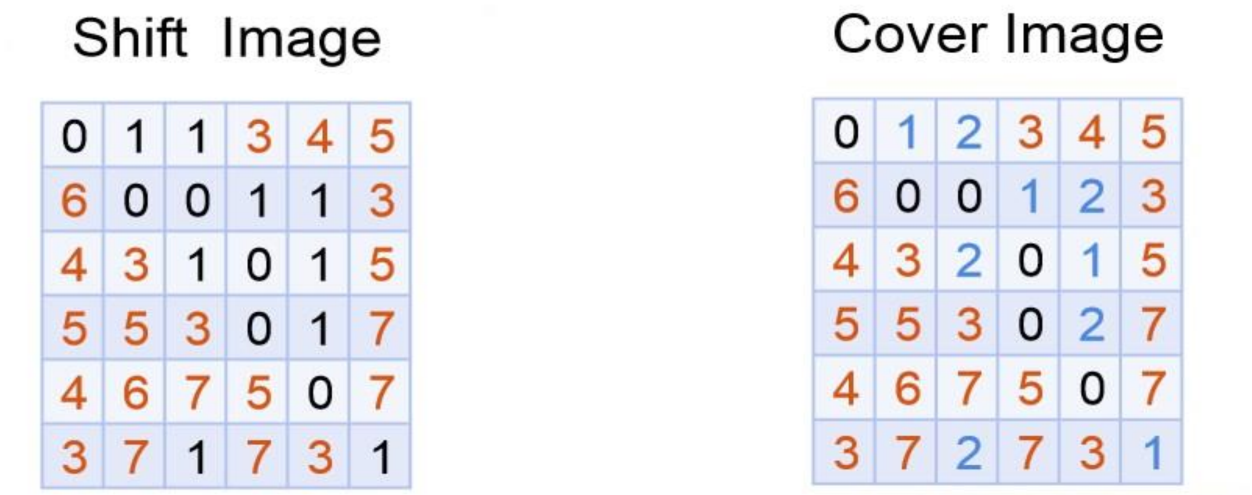


Figure 6.5 Embedding of data [2]

Consider following 3BPP image :

PP = h(1) , PP < ZP

Secret data :010110110



Shift Image

0	1	1	3	4	5
6	0	0	1	1	3
4	3	1	0	1	5
5	5	3	0	1	7
4	6	7	5	0	7
3	7	1	7	3	1

Cover Image

0	1	2	3	4	5
6	0	0	1	2	3
4	3	2	0	1	5
5	5	3	0	2	7
4	6	7	5	0	7
3	7	2	7	3	1

Figure 6.6 Shifting of image example [2]

6.4 Peak Value Insertion

After embedding the secret data into the cover image there is a possibility that the peak value may change. The peak value is required for extracting the secret data from the stego-image. This arises the need of storing the peak value into the stego-image. The last pixel of the cover image is reserved for storing the peak value, hence the data bit cannot be embedded at this pixel position.

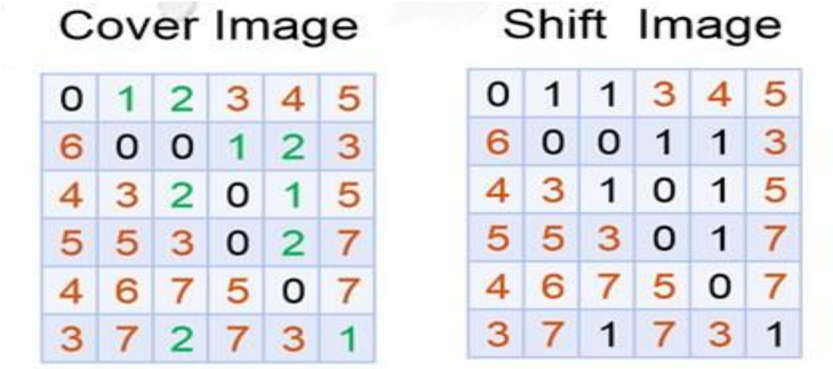
6.5 De-Embedding

To obtain the secret data from inside the cover image de-embedding process is used. The following algorithm is used to perform de-embedding:

- Step 1: Obtain the peak point and the zero point from the stored record.
- Step 2: Scan the stego-image in the same sequential order that was used in the data embedding process. If the pixel value is in the range [PP+1, ZP], decrease the pixel

- value by one to recover its original value. At the same time, a bit "1" is extracted if the pixel value is P+1; a bit "0" is extracted if the pixel value is PP.
- Step 3: Restore the corresponding pixel values if the extracted data include the overhead bookkeeping information, which is stored in Step 2 of the data embedding process.

PP = h(1) , PP < ZP and Secret data :010110110



Cover Image

0	1	2	3	4	5
6	0	0	1	2	3
4	3	2	0	1	5
5	5	3	0	2	7
4	6	7	5	0	7
3	7	2	7	3	1

Shift Image

0	1	1	3	4	5
6	0	0	1	1	3
4	3	1	0	1	5
5	5	3	0	1	7
4	6	7	5	0	7
3	7	1	7	3	1

Figure 6.7 Extraction of data [2]

6.6 AES Decryption

AES decryption is not identical to encryption since steps done in reverse but can define an equivalent inverse cipher with steps as for encryption using inverses of each step with a different key schedule. This works since result is unchanged when swap byte substitution & shift rows swap mix columns & add (tweaked) round key. The algorithm is explained using the flow diagram as follows:

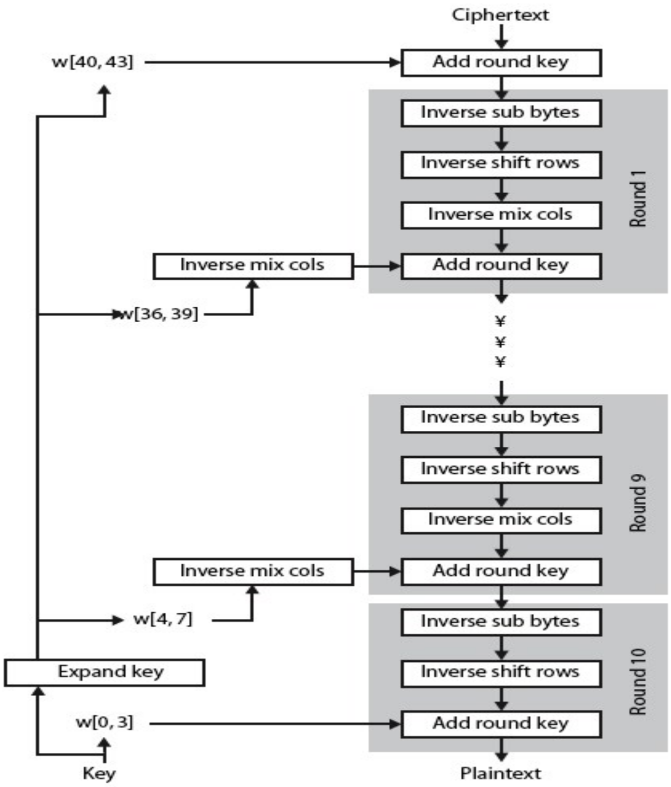
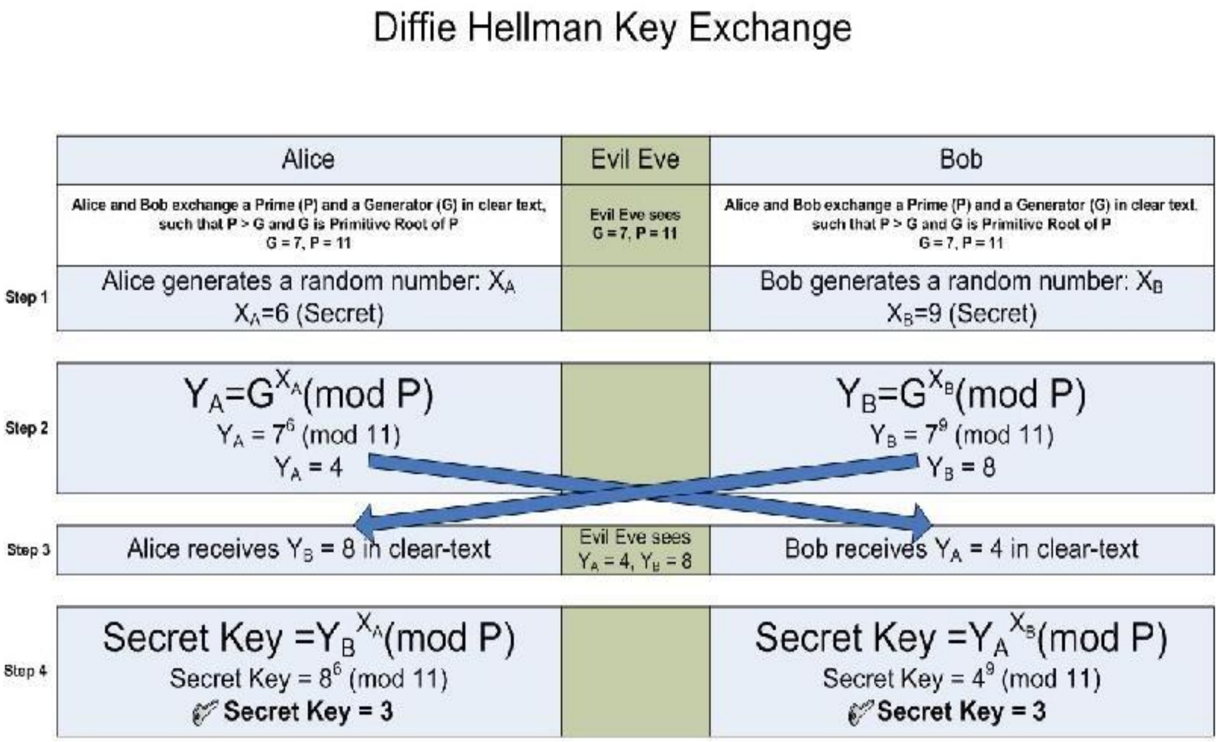


Figure 6.8 AES Decryption Process [1]

6.7 Diffie-Hellman Key Exchange Protocol

The Diffie–Hellman key exchange algorithm solves the following dilemma. Alice and Bob want to share a secret key for use in a symmetric cipher, but their only means of communication is insecure. Every piece of information that they exchange is observed by their adversary Eve. How is it possible for Alice and Bob to share a key without making it available to Eve? At first glance it appears that Alice and Bob face an impossible task. It was a brilliant insight of Diffie and Hellman that the difficulty of the discrete logarithm problem for F∗ p provides a possible solution. The first step is for Alice and Bob to agree on a large prime p and a nonzero integer g modulo p. Alice and Bob make the values of p and g public knowledge; for example, they might post the values on their web sites, so Eve knows them, too. It is best if they choose g such that its order in F∗ p is a large prime.



Diffie Hellman Key Exchange

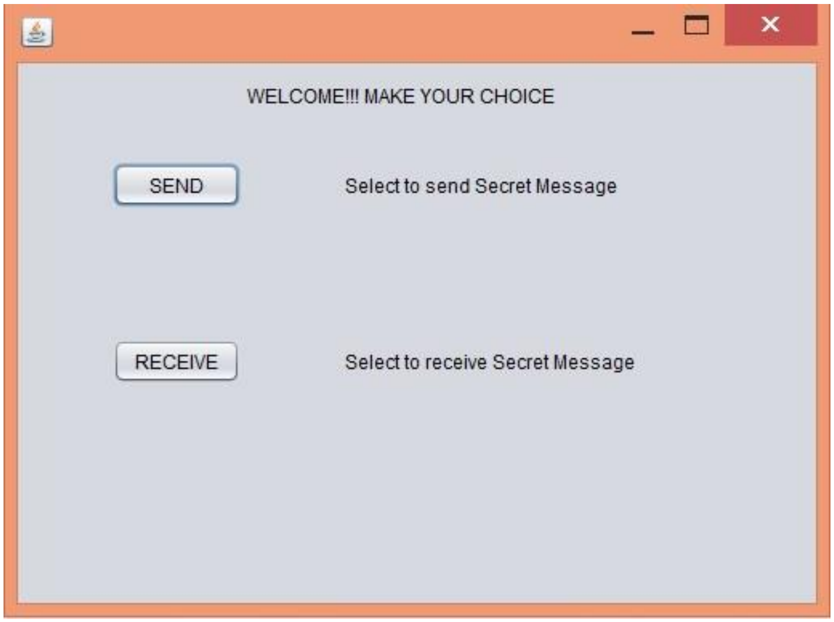
	Alice	Evil Eve	Bob
	Alice and Bob exchange a Prime (P) and a Generator (G) in clear text, such that P > G and G is Primitive Root of P G = 7, P = 11	Evil Eve sees G = 7, P = 11	Alice and Bob exchange a Prime (P) and a Generator (G) in clear text. such that P > G and G is Primitive Root of P G = 7, P = 11
Step 1	Alice generates a random number: X_A X_A=6 (Secret)		Bob generates a random number: X_B X_B=9 (Secret)
Step 2	$Y_A=G^{X_A}(\text{mod } P)$ $Y_A = 7^6$ (mod 11) $Y_A = 4$		$Y_B=G^{X_B}(\text{mod } P)$ $Y_B = 7^9$ (mod 11) $Y_B = 8$
Step 3	Alice receives Y_B = 8 in clear-text	Evil Eve sees $Y_A = 4$, $Y_B = 8$	Bob receives Y_A = 4 in clear-text
Step 4	Secret Key =$Y_B{}^{X_A}$(mod P) Secret Key = 8^6 (mod 11) **Secret Key = 3**		Secret Key =$Y_A{}^{X_B}$(mod P) Secret Key = 4^9 (mod 11) **Secret Key = 3**

Figure 6.9 Diffie- Hellman Key Exchange Protocol

The next step is for Alice to pick a secret integer a that she does not reveal to anyone, while at the same time Bob picks an integer b that he keeps secret. Bob and Alice use their secret integers to computeA ≡ ga (mod p) | {z } Alice computes thisand B ≡ gb (mod p) | {z } Bob computes this. They next exchange these computed values, Alice sends A to Bob and Bob sends B to Alice. Note that Eve gets to see the values of A and B, since they are sent over the insecure communication channel. Finally, Bob and Alice again use their secret integers to compute A0 ≡ Ba (mod p) | {z } Alice computes this and B0 ≡ Ab (mod p) | {z } Bob computes this . The values that they compute, A0 and B0 respectively, are actually the same, since A0 ≡ Ba ≡ (gb)a ≡ gab ≡ (ga)b ≡ Ab ≡ B0 (mod p). This common value is their exchanged key.

6.8 Snapshots

6.8.1 Selection of send or receive



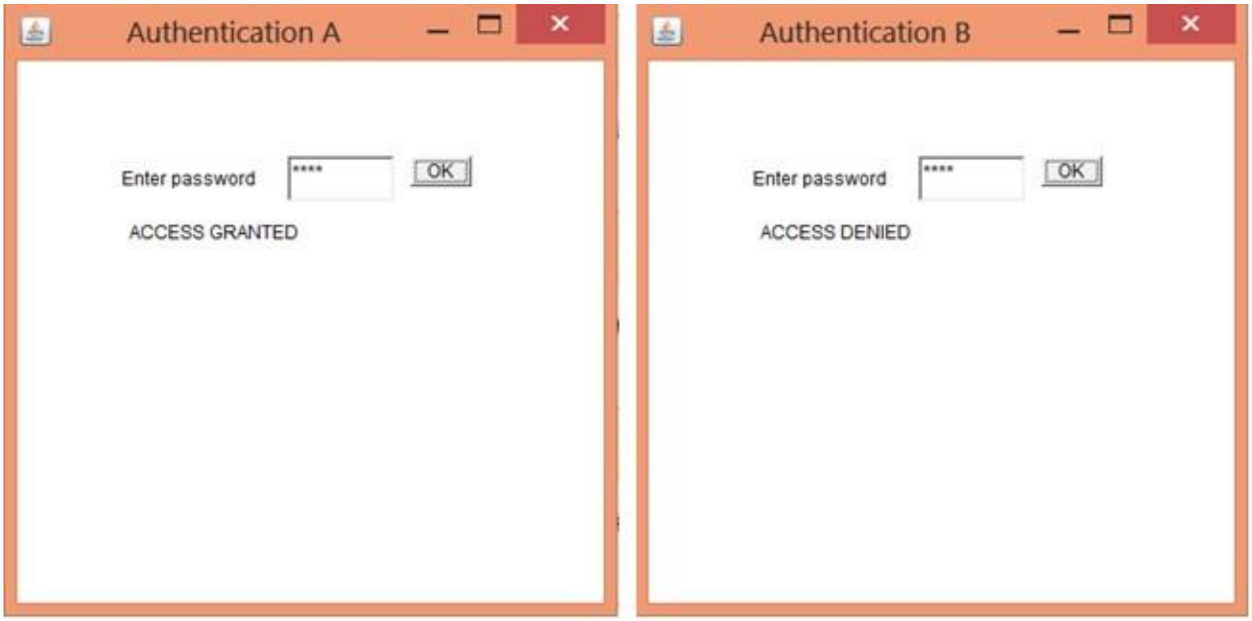
WELCOME!!! MAKE YOUR CHOICE

Select to send Secret Message

Select to receive Secret Message

Figure 6.10 Snapshot1

6.8.2 After selection user needs to authenticate.



Authentication A

Enter password ****

ACCESS GRANTED

Authentication B

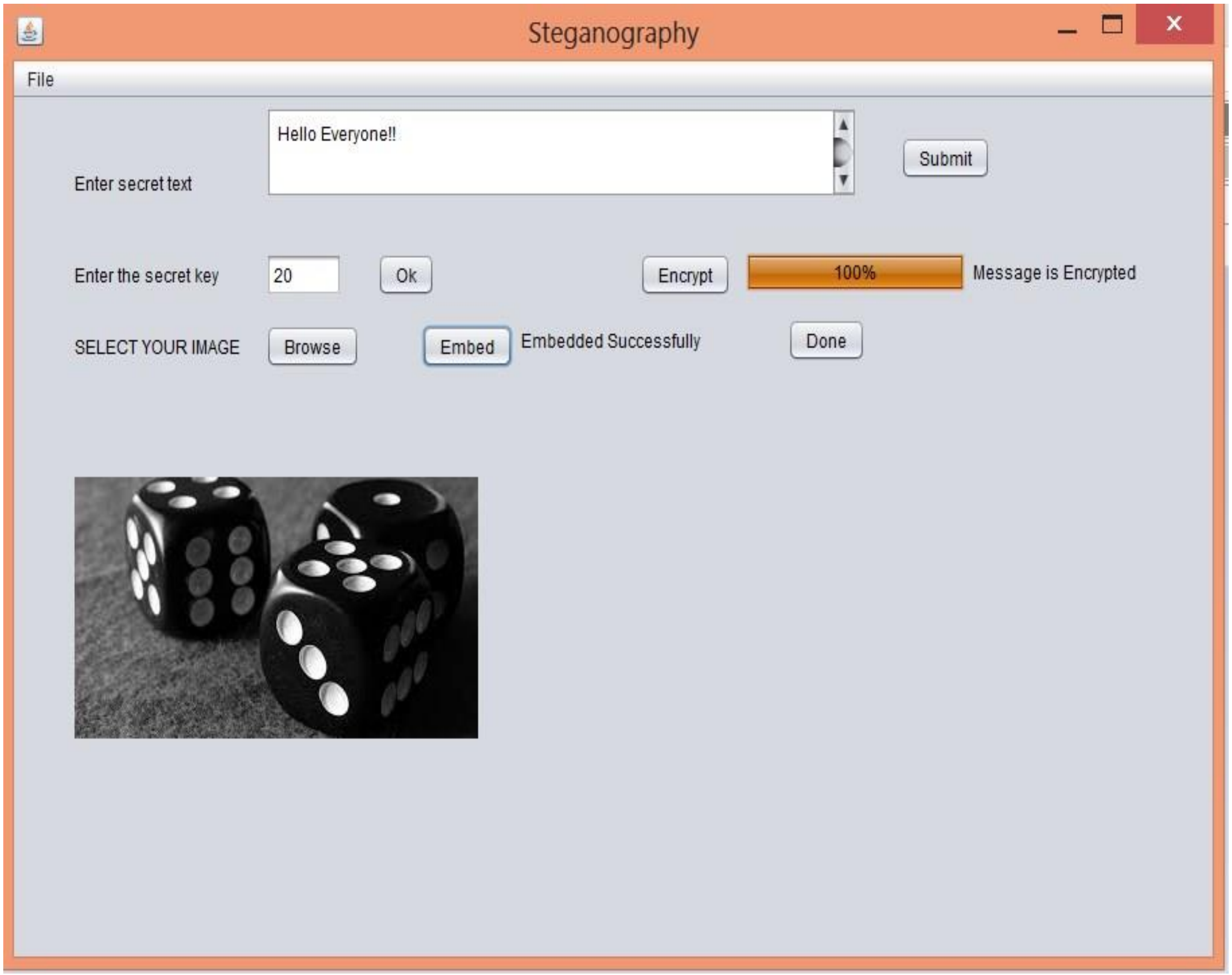
Enter password ****

ACCESS DENIED

Figure 6.11 Snapshot 2

6.8.3 User specifies the secret message and cover image

Secret message is encrypted using the exchanged key which is hidden inside a cover image and then embedding is done



Enter secret text: Hello Everyone!!

Enter the secret key: 20

Message is Encrypted

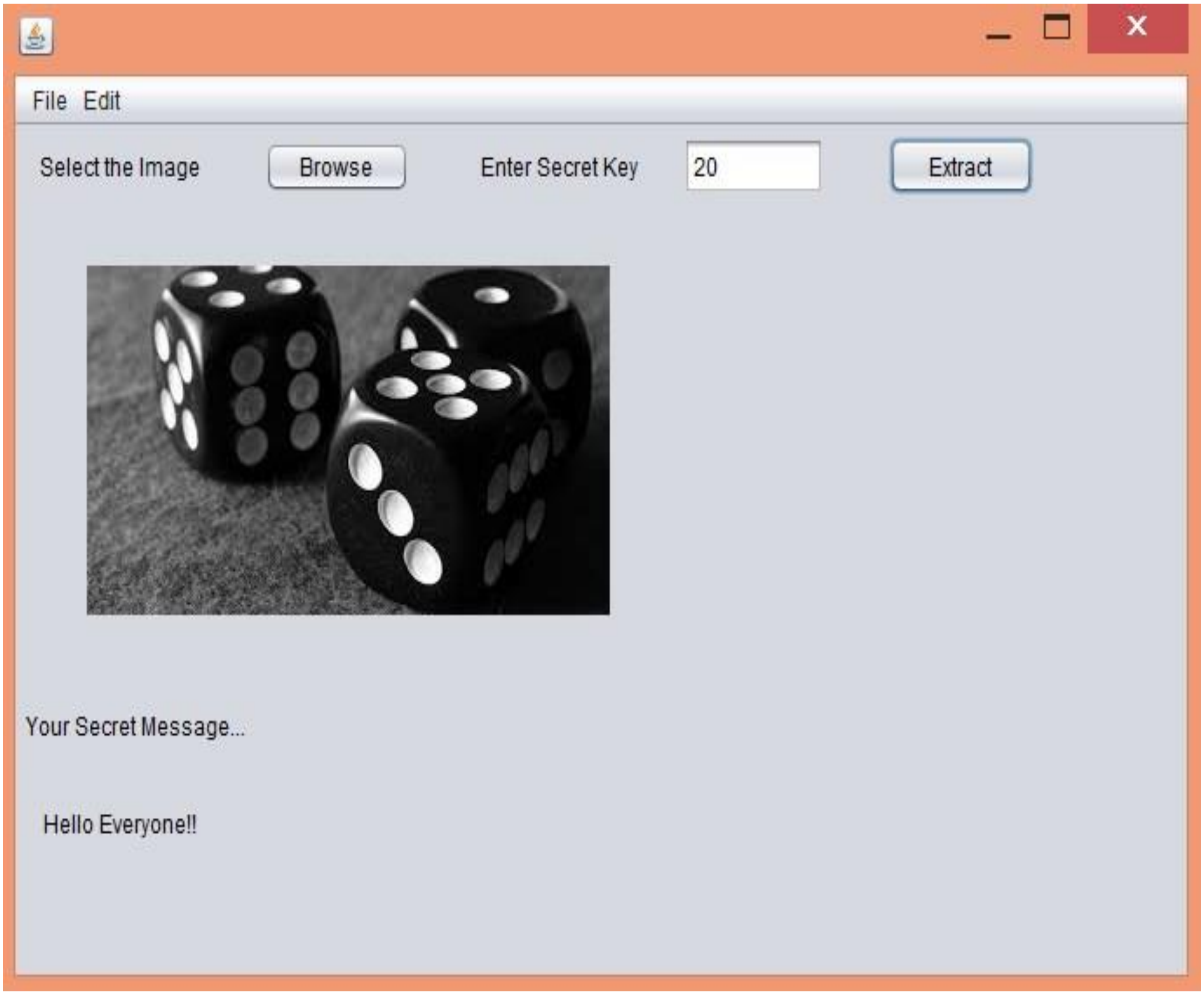
SELECT YOUR IMAGE

Embedded Successfully

Figure 6.12 Snapshot 3

6.8.4 User selects the receive option

User needs to authenticate to proceed and following screen appears, where receiver selects the received Stego- image, enters the secret key and extraction is performed



20

Your Secret Message...

Hello Everyone!!

Figure 6.13 Snapshot 4

Chapter 7
Testing

Testing is done to identify the efficiency and accuracy of the system developed. Testing is done by taking into consideration various test cases and testing them under different conditions and circumstances

Testing is performed on five test images used as a cover media, each having different embedding capacity. The application successfully embedded and de-embedded the secret message and the PSNR value for each case was calculated and obtained above 49 dB .The result is as follows:

Table 7.1 Test Cases

TEST CASE ID	OBJECTIVE	STEPS	INPUT DESCRIPTION	EXPECTED OUTPUT	ACTUAL OUTPUT	RESULT	REMARK
1. Car	Embedding	1.Encryption 2.Embedding 3.Extraction 4.Decryption	1.Secret message 2.Cover Gray scale Image			50.247 PSNR	Successful
2. Dice	Embedding	1.Encryption 2.Embedding 3.Extraction 4.Decryption	1.Secret message 2.Cover Gray scale Image			49.706 PSNR	Successful
3. Ship	Embedding	1.Encryption 2.Embedding 3.Extraction 4.Decryption	1.Secret message 2.Cover Gray scale Image			49.451 PSNR	Successful
4. Lion	Embedding	1.Encryption 2.Embedding 3.Extraction 4.Decryption	1.Secret message 2.Cover Gray scale Image			55.987 PSNR	Successful
5. Spiral	Embedding	1.Encryption 2.Embedding 3.Extraction 4.Decryption	1.Secret message 2.Cover Gray scale Image			59.927 PSNR	Successful

Chapter 8
Results and Analysis

The secret message is successfully embedded into the grey scale cover image. The Stego image has PSNR value above 48 dB and have no visible physical changes and the secret message is obtained successfully.

The results of the following Case is:

1. The Secret Message “Hello Everyone!!!” is entered by the user and the Encryption Key chosen is 20

2. The Cover Image Selected is

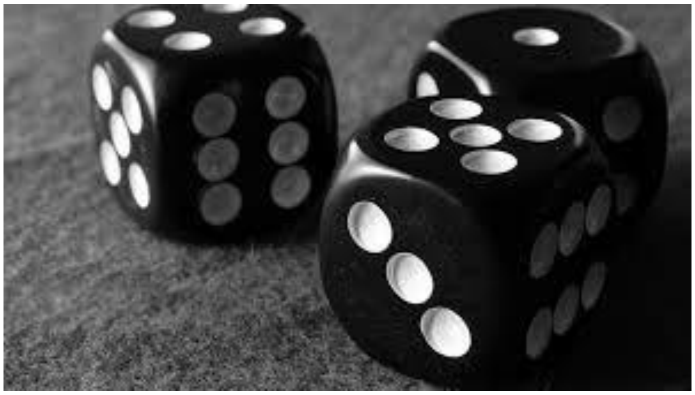


Figure 8.1 Cover Image

The Peak point and the Zero point of the selected cover image is as follows

- Peak Point: 0 and number of pixels: 1963
- Zero Point: 173 and number of pixels: 9
- Thus, capacity for embedding is 1963 bits.

3. Encryption of the input secret text is performed using the secret key, and the cipher is obtained
    - Input text: Hello Everyone!!!
    - Encrypted Text : DRkP6sd9ckQ6g7WLrdO9SLF/kUrjn6zuGdifGsQu3s0=

4. The Cipher text thus obtained is converted to Bit Stream, the length of the Stream can be computed as Bit-Stream Length=8* number of characters in the encrypted text

    For the current scenario Bit-Stream Length=8*44=352

    - Bit-Stream:
      0100010001010010011010110101000000110110011100110110010000111001011000110110101101010001001101100110011100110111010101110100110001110010011001000100111100111001010100110100110001000110001011110110101101010101011100100110101001101110001101100111101001110101010001110110010001101001011001100100011101110011010100010111010100110011011100110011000000000111101
    - Bit-Stream Length:352

5. The Bit-Stream obtained is embedded inside the cover image and following Stego-Image is obtained

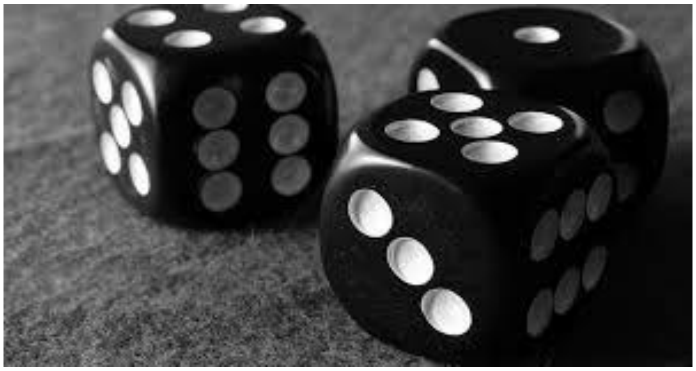
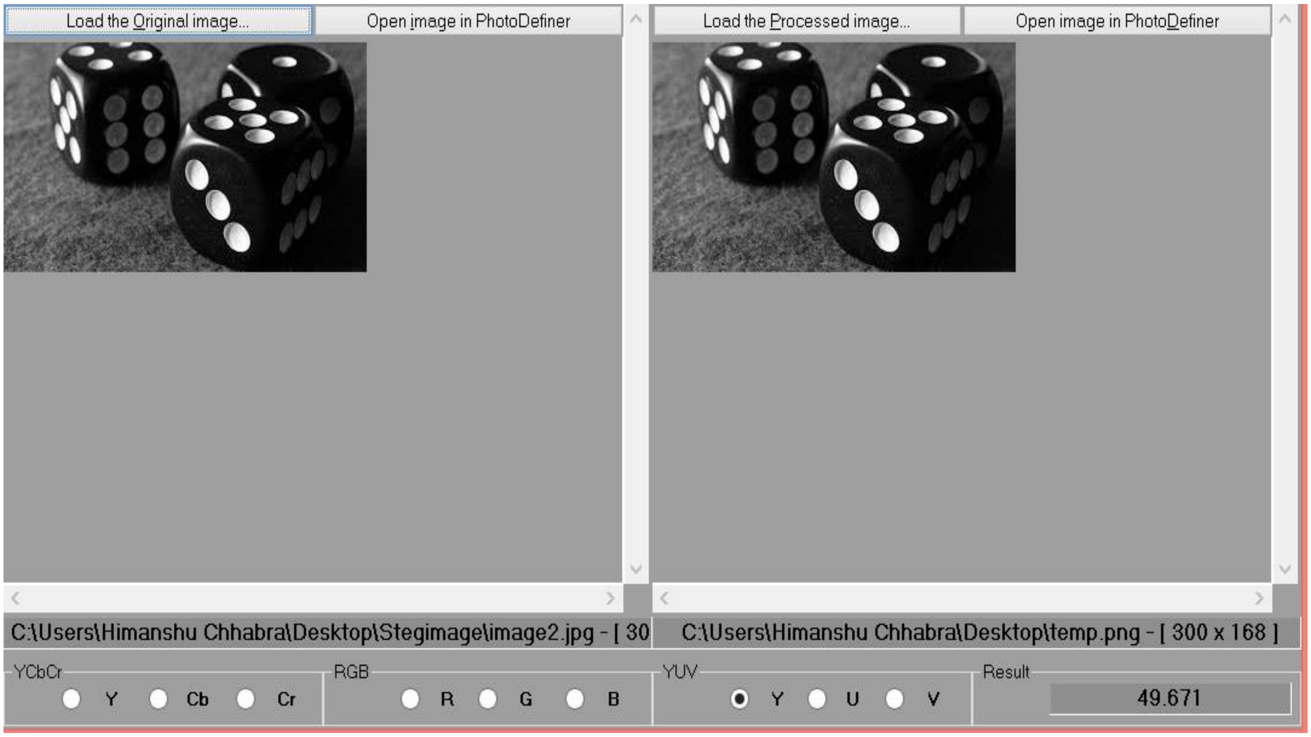


Figure 8.2 Stego Image (Peak Signal to Noise ratio (PSNR): 49.671)

6. Comparison Between the Cover Image and Stego-Image


Load the Original image...
Open image in PhotoDefiner
Load the Processed image...
Open image in PhotoDefiner

C:\Users\Himanshu Chhabra\Desktop\Stegimage\image2.jpg - [30

C:\Users\Himanshu Chhabra\Desktop\temp.png - [300 x 168]

YCbCr: ○ Y ○ Cb ○ Cr

RGB: ○ R ○ G ○ B

YUV: ● Y ○ U ○ V

Result: 49.671

Figure 8.3 Comparison of Orignal and Stego Image

7. The Stego Image is sent to the Receiver, the image can withstand Zip compressions and Email compressions. Thus it can be sent by mail to the intended recipient. If the maximum capacity of the cover image is reached, user can make use of same cover image and embed the remaining message with same encryption key and can zip the obtained Stego images into a file/folder and send in one short.

8. The Received Stego image undergoes the extraction process and the following Bit-Steam is extracted and the stream is converted back into its character form and the Cipher text is obtained

   - Bit-Stream:
     010001000101001001101011010100000011011001110011011001000011100101100
     01101101011010100010011011001100111001101110101011101001100011100100 1
     10010001001111001110010101001101001100010001100010111101101011010101 0
     10111001001101010011011100011011001111010011101010100011101100100011 0

- 1001011001100100011101110011010100010111010100110011011100110011000000111101
- Bit-Stream Length: 352
- Cipher Text:DRkP6sd9ckQ6g7WLrdO9SLF/kUrjn6zuGdifGsQu3s0=

9. The Decryption Key entered by the user must match with the Encryption Key if it does decryption of the cipher text takes place and the secret message is obtained, else the decryption doesn’t take place and user can try again.
    - Secret Message: Hello Everyone!!!

10. The Stego Image can be used again as a cover image and new message can be embedded into it, which means the image is Reusable.

11. Computation time for embedding and de-embedding are as follows

    Embedding Time: 4 sec

    The time is inclusive of encrypting the secret text, its conversion into binary form, reading pixels of the cover image, processing it and forming the Stego image

    De-Embedding Time: 4 sec

    The time is inclusive of reading the Stego image, extracting bits, conversion to character form and decryption.

12. The encryption of the secret text adds to the security and robustness of the application.

Chapter 9
Conclusion and Future Scope

The algorithm used can successfully embed as well as extract the secret text from the cover image. The experiments with the test image show that the algorithm yields images with PSNR not less than 48db.The Algorithm is simple, has uniformity and execution time is short. This technique is useful in various domains like military where confidentiality and Secrecy is at most important. The procedure used yields file having png format. The stego images can withstand compressions of email and zip compression format. Additionally encryption of the secret text adds up to the security of the algorithm. Its overall performance is good.

The embedding capacity can be increased by selecting multiple pair of peak and zero points. The embedding procedure remains the same with the following changes:

1. As a record of occurrences of each pixel value is maintained, consider for a three level embedding process a pair of three PP and ZP such that they do not overlap are selected from the record.
2. If pp1,pp2,pp3 are three peak points and zp1,zp2,zp3 are three zero points such that pp1<pp2<pp3 and zp3<zp2<zp1 then the recommended order of using the pairs will be [pp1,zp1] , [pp2,zp2] , [pp3,zp3], there are no restrictions on zero points, they can form a

3. pair with any peak point p1, p2 or p3 but the order of using them must include p1 first followed by p2 and then p3.This is necessary to avoid loss of data
4. For the first iteration it is as explained before, for remaining iterations concatenate the bit stream of the secret data of the previous iteration with eight “0” followed by the PP of the next iteration in its binary 8 bit equivalent form where the eight “0” separates the secret data from the PP of the next iteration.
5.
6. During the extraction process the eight bits followed by the eight “0” at the end of the extracted bit streams will be used as the peak value (PP) for the next iteration.

The above procedure is continued until the extracted bit stream does not contain a stream of eight “0” at the end.

Appendix

This section contains theory in regards to Secure Data Hiding with Advanced Encryption Standard.

A

Authentication: This attribute is used to verify the intended user.

C

Cipher: This attribute refers to an encrypted form of text.

D

Decryption: This attribute decrypts the encrypted text.

De-embedding: This attribute refers to obtaining the text from the image.

E

Encryption: This attribute refers to converting the message into cipher text.

Embedding: This attribute refers to hiding the text in the cover image.

P

Peak Point: This attribute refers to the intensity value which maximum number of pixels possess.

PSNR: this attribute specifies the ratio of the original image to the amount of noise in the processed image.

R

Receiver: This attribute refers to the person to whom the stego image is sent.

S

Sender: This attribute refers to the person that sends the stego image.

Stego Image: This attribute refers to the image in which the message is hidden.

Z

Zero Point: This attribute refers to the intensity value which minimum number of pixels possess.

References

[1] Zhicheng Ni, Yun-Qing Shi, Nirwan Ansari, and Wei Su, “Reversible Data Hiding” in IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 16, NO. 3, MARCH 2006.

[2] Y.-C. Li, C.-M. Yeh and C.-C. Chang, “Data hiding based on the similarity between neighboring pixels with reversibility” in Digital Signal Processing, vol. 20, no. 4, pp. 1116–1128, 2010.

[3] J.Anita Christaline, D. Vaishali , “IMAGE STEGANOGRAPHIC TECHNIQUES WITH IMPROVED EMBEDDING CAPACITY AND ROBUSTNESS” in IEEE-International Conference on Recent Trends in Information Technology, ICRTIT 2011 IEEE MIT, Anna University, Chennai, 978-1-4577-0590-8/11, June 3-5, 2011.

[4] Jagbir Singh, Savina Bansal and R.K. Bansal, “Performance Analysis of Data Hiding Using Adjacent Pixel Difference Technique” in International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 9, September 2013

[5] IEEE-International Conference on Recent Trends in Information Technology, ICRTIT 2011 9678-1-4577- 0590-8/11/$26.00 ©2011 IEEE MIT, Anna University, Chennai. June 3-5, 2011. IMAGE STEGANOGRAPHIC TECHNIQUES WITH IMPROVED EMBEDDING CAPACITY AND ROBUSTNESS

[6] International Journal of Computer Science and Security, (IJCSS), Volume (4): Issue (6).

A NOVEL TECHNIQUE FOR IMAGE STEGANOGRAPHY BASED ON DWT AND HUFFMAN ENCODING.

[7] Journal of Computer Science, 9 (3): 277-284, 2013 ISSN 1549-3636 © 2013 Science Publications doi:10.3844/jcssp.2013.277.284 Published Online 9 (3) 2013 277 Science

Publications JCS. A HIGH SECURE AND ROBUST IMAGE STEGANOGRAPHY USING DUAL WAVELET AND BLENDING MODEL

[8] I. J. Computer Network and Information Security, 2012, 7, 27-40 Published Online July 2012 in MECS. A ROBUST IMAGE STEGANOGRAPHIC USING DWT DIFFERENCE MODULATION (DWTDM)

[9] International Journal of Power Control Signal and Computation (IJPCSC) Vol. 4. No.2. pp.102 -108 April - June 2012 ISSN: 0976-268X www.ijcns.com. ROBUST DISCRETE WAVELET TRANSFORM BASED STEGANOGRAPHY.

[10] William Stallings, “Cryptography and Network Security” 5/e, Chapter 5 –“Advanced Encryption Standard”.

Acknowledgements

The project on “Secure Data Hiding with Advanced Encryption Standard” is an outcome of the guidance, moral support and devotion bestowed upon us throughout our work.

For this, we acknowledge and express our profound sense of gratitude and thanks to everybody who have been a source of inspiration during the seminar preparation.

We are thankful to our Faculty professors who guided us throughout the project preparation and provided help whenever we needed. We would also like to thank our principal. We would like to extend our sincere thanks to all of them. We are highly indebted to Mrs. Rupali Kale and Mrs. Vidyulata Devmane and their guidance and constant supervision.

We also thank each and every one who has helped us directly and indirectly in the preparation of this seminar report.

-- --

(Himanshu Chhabra) (Dhiraj Thakur)

(Riddhi Thacker)

Date: