CHAPTER 1

INTRODUCTION

Since the rise of the Internet one of the most important factors of information technology and communication has been the security of information. Cryptography was created as a technique for securing the secrecy of communication and many different methods have been developed to encrypt and decrypt data in order to keep the message secret. Unfortunately it is sometimes not enough to keep the contents of a message secret, it may also be necessary to keep the existence of the message secret. The technique used to implement this, is called steganography. Steganography is the art and science of invisible communication. This is accomplished through hiding information in other information, thus hiding the existence of the communicated information. The word steganography is derived from the Greek words “stegos” meaning “cover” and “grafia” meaning “writing” defining it as “covered writing”. In image steganography the information is hidden exclusively in images. The idea and practice of hiding information has a longhistory.

In Histories the Greek historian Herodotus writes of a nobleman, Histaeus, who needed to communicate with his son-in-law in Greece. He shaved the head of one of his most trusted slaves and tattooed the message onto the slave’sscalp. When the slave’shair grewback the slave was dispatched with the hidden message. In the Second World Warthe Microdot technique was developed by the Germans. Information, especially photographs, was reduced in size until it was the size of a typed period. Extremely difficult to detect, a normal cover message was sent over an insecure channel with one of the periods on the paper containing hidden information. Today steganography is mostly used on computers with digital data being the carriers and networks being the high speed delivery channels. Steganography differs from cryptography in the sense that where cryptography focuses on keeping the existence of a message secret.

Steganography and cryptography are both ways to protect information from unwanted parties but neither technology alone is perfect and can be compromised. Once the presence of hidden information is revealed or even suspected, the purpose of steganography is partly defeated. The strength of steganography can thus be amplified by combining it with cryptography. Two other technologies that are closely related to steganography are watermarking and finger printing. These technologies are mainly concerned with the protection of intellectual property, thus the algorithms have different requirements than steganography. These requirements of a good steganographic algorithm will be discussed below. In watermarking all of the instances of an object are “marked” in the same way. The kind of information hidden in objects when using watermarking is usually a signature to signify origin or ownership for the purpose of copyright protection .With fingerprinting on the other hand, different, unique marks are embedded in distinct copies of the carrier object that are supplied to differentcustomers.

This enables the intellectual property owner to identify customers who break their licensing agreement by supplying the property to third parties . In watermarking and fingerprinting the fact that information is hidden inside the files may be public knowledge sometimes it may even be visible while in steganography the imperceptibility of the information is crucial. Aattack on a steganographic system consists of an adversary observing that there is information hidden inside a file, while a successful attack on a watermarking system would not be to detect mark,but to removeit.

Overview of Steganography:

To provide an overview of steganography, terms and concepts should first be explained.An overview of the different kinds of steganography is given at a later stage.Steganography concepts Although steganography is an ancient subject, the modern formulation of it is often given in terms of the prisoner's problem proposed by Simmons ,where two inmates wish to communicate in secret to hatch an escape plan.

All of their communication passes through a warden who will throw them in solitary confinement should she suspect any covert communication. The warden, who is free to examine all communication exchanged between the inmates, can either be passive or active. A passive warden simply examines the communication to try and determine if itpotentiallycontains secret information. If she suspects a communication to contain hidden information, a passive warden takes note of the detected covert communication, reports this to some outside party and lets the message through without blocking it. An active warden, on the other hand, will try to alter the communication with the suspected hidden information deliberately, in order to remove theinformation

Different kinds of steganography:

Almost all digital file formats can be used for steganography, but the formats that are more suitable are those with a high degree of redundancy. Redundancy can be defined as the bits of an object that provide accuracy far greater than necessary for the object's use and display. The redundant bits of an object are those bits that can be altered without the alteration being detected easily. Image and audio files especially comply with this requirement, while research has also uncovered other file formats that can be used for information hiding.

Stegnography is divided into 3 types:

- Image
- Audio

- Video

Hiding information in text is historically the most important method of steganography. An obvious method was to hide a secret message in every nth letter of every word of a text message. It is only since the beginning in the Internet and all the different digital file formats that is has decreased in importance. Text steganography using digital files is not used very often since text files have a very small amount of redundant data. Given the proliferation of digital images, especially on the Internet, and given the large amount of redundant bits present in the digital representation of an image, images are the most popular cover objects for steganography.

This paper will focus on hiding information in images in the next sections. To hide information in audio files similar techniques are used as for image files. One different technique unique to audio steganography is masking, which exploits the properties of the human ear to hide information unnoticeably. A faint, but audible, sound becomes inaudible in the presence of another louder audible sound. This property creates a channel in which to hide information. Although nearly equal to images in steganographic potential, the larger size of meaningful audio files makes them less popular to use than images. The term protocol steganography refers to the technique of embedding information within messages and network control protocols used in network transmission . In the layers of the OSI network model there exist covert channels where steganography can be used. An example of where information can be hidden is in the header of a TCP/IP packet in some fields that are either optional rare never used.A paper by Kundur provides more information on this.

1.2 ExistingSystem:

However, we find that in most existing approaches, the choice of embedding positions within a cover image mainly depends on a pseudorandom number generator without considering the relationship between the image content itself and the size of the secret message.

We find that the existing PVD-based approaches cannot make full use of edge information for data hiding, and they are also poor at resisting some statistical analyses.

1.3 Proposed System:

We expand the LSB matching revisited image steganography and propose an edge adaptive scheme which can select the embedding regions according to the size of secret message and the difference between two consecutive pixels in the coverimage.

For lower embedding rates, only sharper edge regions are used while keeping the other smoother regions as they are. When the embedding rate increases, more edge regions can be released adaptively for data hiding by adjusting just a fewparameters

1.4 Objective:

The project is carried out with the following objectives:

- To hide the message or a secret data into an image this acts as a cover medium using LSB technique and pseudo random technique.
- The primary motivation of my current work is to increase PSNR of the stegoimage (peak signal to noiseratio).

CHAPTER 2

LITERATURE SURVEY

The term steganography came into use in 1500s after the appearance of Trithemius book on the subject Steganographia. Past: The word Steganography technically means covered or hidden writing. Its ancient origins can be traced back to 440 BC. Although the term steganography was only coined at the end of the 15th century, the use of steganography dates back several millennia. In ancient times, messages were hidden on the back of wax writing tables, written on the stomachs of rabbits, or tattooed on the scalp of slaves. Invisible ink has been in use for centuries for fun by children and students and for serious undercover work by spies and terrorists.

Present :The majority of today's steganographic systems uses multimedia objects like image, audio, video etc as cover media because people often transmit digital pictures over email and other Internet communication. Modern steganography uses the opportunity of hiding information into digital multimedia files and also at the net- work packet level.

Hiding information into a medium requires following elements.

- The cover medium(C) that will hold the secret message.
- The secret message (M), may be plain text, digital image file or any type of data
- The stegonographic technique
- A stegokey (K) may be used to hide and unhide the message.

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

- Text Steganography
- Image Steganography
- Audio Steganography
- Video Steganography
- Protocol Steganography

Text steganography: Text steganography hiding information in text file is the most common method of steganography. The method was to hide a secret message into a text message. After coming of Internet and different type of digital file formats it has decreased in importance. Text stenography using digital files is not used very often because the text files have a very small amount of excess data.

Image steganography: Images are used as the popular cover medium for steganography. A message is embedded in a digital image using an embedding algorithm, using the secret key. The resulting stego imageis send to the receiver. On the other side, it is processed by the extraction algorithm using the same key. During the transmission of stego image unauthenticated persons can only notice the transmission of an image but can't see the existence of the hidden message.

Audio steganography: Audio steganography is concerned with embedding information in an innocuous cover speech in a secure and robust manner. Communication and transmission security and robustness are essential for transmitting vital information to intended sources while denying access to unauthorized persons. An audible, sound can be inaudible in the presence of another louder audible sound .This property allows to select the channel in which to hide information. Existing audio steganography software can embed messages in WAV and MP3 sound files. The list of methods that are commonly used for audio steganography are listed and discussed below.

Video steganography: Video Steganography is a technique to hide any kind of files in any extension into a carrying Video file.

Protocol steganography : Protocol steganography the term protocol steganography is to embed- ding information within network protocols such as TCP/IP. We hide information in the header of a TCP/IP packet in some fields that can be either optional or are never used.

CHAPTER 3

SYSTEM REQUIREMENTS

3.1 Hardware Requirements:

System	: Pentium IV 2.4GHz.
Hard Disk	: 4GB.
Floppy Drive	: 1.44MB.
Monitor	: 15 VGAColour.
Mouse	:Logitech.
RAM	: 256MB.

3.2 Software Requirements:

Operating system	: – Windows XP Professional.
Front End	: – Visual Studio.Net2005
Coding Language	: –python
Back End	: – SQL2000.

CHAPTER 4

SYSTEM STUDY

4.1 FEASIBILITY STUDY:

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential. Three key considerations involved in the feasibility analysis are

- ECONOMICALFEASIBILITY
- TECHNICALFEASIBILITY
- SOCIALFEASIBILITY

4.1.1 ECONOMICALFEASIBILITY:

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

4.1.2 TECHNICALFEASIBILITY:

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being place do the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

4.1.3 SOCIAL FEASIBILITY:

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

CHAPTER 5

SYSTEM ARCHITECTURE

SYSTEM ARCHITECTURE :

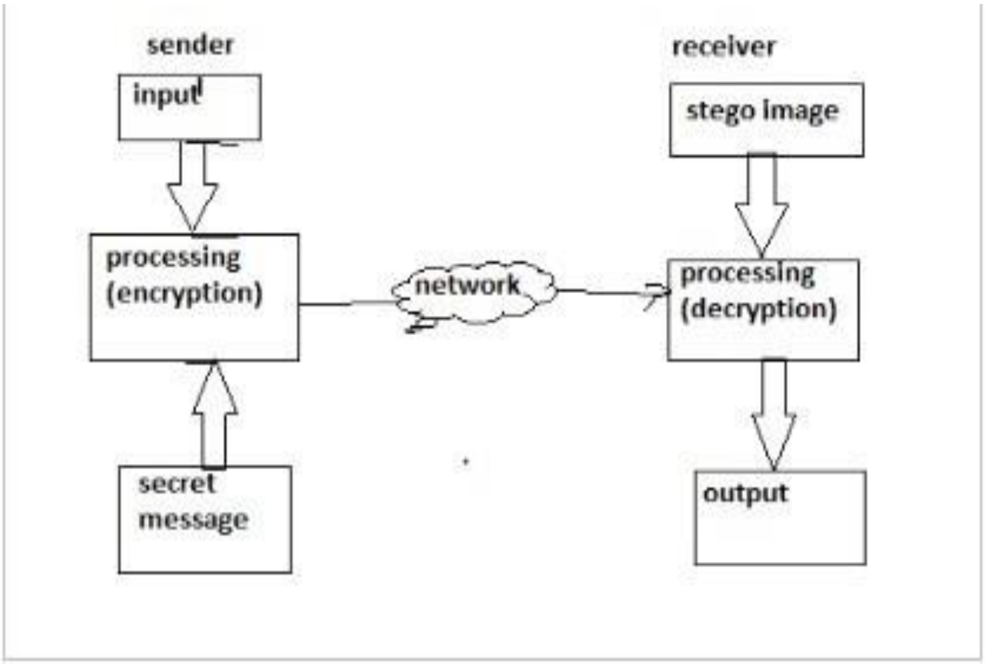


Fig: 5.1: System Architecture for Image Steganography

CHAPTER 6

MODULES

6.1 MODULES:

Steg module: In this module user need to write any secret information in text area provided and need to select an image file to which user wants to append the secret information text.

Unsteg module: In this module the encrypted image will be selected and stegnography application will start decrypting it with each and every pixel of the image and displays the output i.e., secret text information.

CHAPTER 7

TECHNOLOGY DESCRIPTION

Python:

It is an open source programming language that was made to be easy-to-read and powerful. A Dutch programmer named Guido van Rossum made Python in 1991. He named it after the television show Monty Python's Flying Circus. Many Python examples and tutorials include jokes from the show.

Python is an interpreted language. Interpreted languages do not need to be compiled to run. A program called an interpreter runs Python code on almost any kind of computer. This means that a programmer can change the code and quickly see the results. This also means Python is slower than a compiled language like C, because it is not running machine code directly.

Python is a good programming language for beginners. It is a high-level language, which means a programmer can focus on what to do instead of how to do it. Writing programs in Python takes less time than in some other languages.

Drew inspiration from other programming languages like C, C++, Java, Perl, and Lisp.

Python's developers strive to avoid premature optimization. Additionally, they reject patches to non-critical parts of the Python reference implementation that would provide improvements on speed. When speed is important, a Python programmer can move time-critical functions to extension modules written in languages such as C or Py, a just-in-time compiler. Python is also available. It translates a Python script into C and makes direct C-level API calls into the Python interpreter.

Keeping Python fun to use is an important goal of Python's developers. It reflects in the language's name, a tribute to the British comedy group Monty Python. On occasions, they are playful approaches to tutorials and reference materials, such as referring to spam and eggs instead of the standard foo and bar.

Contents:

- Python use
- Syntax
- Example
- References
- Other websites

Python Use:

Python is used by hundreds of thousands of programmers and is used in many places. Sometimes only Python code is used for a program, but most of the time it is used to do simple jobs while another programming language is used to do more complicated tasks.

Its standard library is made up of many functions that come with Python when it is installed. On the Internet there are many other libraries available that make it possible for the Python language to do more things. These libraries make it a powerful language; it can do many different things.

Some things that Python is often used for are:

- Web development
- Scientific programming
- Desktop GUIs applications
- Network programming

- Game programming.

Syntax:

Python has a very easy-to-read syntax. Some of Python's syntax comes from C, because that is the language that Python was written in. But Python uses whitespace to delimit code: spaces or tabs are used to organize code into groups. This is different from C. In C, there is a semicolon at the end of each line and curly braces ({}) are used to group code. Using whitespace to delimit code makes Python a very easy-to-read language.

Statements and control flow:

Python's statements include:

- The assignment statement, or the = sign. In Python, the statement x = 2 means that the name x is bound to the integer 2. Names can be rebound to many different types in Python, which is why Python is a dynamically typed language.
- The if statement, which runs a block of code if certain conditions are met, along with else and elif (a contraction of else if from other programming languages). The elif statement runs a block of code if the previous conditions are not met, but the conditions for the elif statement are met. The else statement runs a block of code if none of the previous conditions are met.
- The for statement, which iterates over an iterable object such as a list and binds each element of that object to a variable to use in that block of code, which creates a for loop.
- The while statement, which runs a block of code as long as certain conditions are met, which creates a while loop.
- The def statement, which defines a function or method.

- The pass statement, which means "do nothing."
- The class statement, which allows the user to create their own type of objects like what integers and strings are.
- The import statement, which imports Python files for use in the user's code.
- The print statement, which outputs various things to the console.

VMware Workstation:

It is a hosted hypervisor that runs on x64 versions of Windows and Linux operating systems[4] (an x86 version of earlier releases was available);[3] it enables users to set up virtual machines (VMs) on a single physical machine, and use them simultaneously along with the actual machine. Each virtual machine can execute its own operating system, including versions of Microsoft Windows, Linux, BSD, and MS-DOS. VMware Workstation is developed and sold by VMware, Inc., a division of Dell Technologies. There is a free-of-charge version, VMware .

Workstation Player, for non-commercial use. An operating systems license is needed to use proprietary ones such as Windows. Ready-made Linux VMs set up for different purposes are available from several sources.

VMware Workstation supports bridging existing host network adapters and sharing physical disk drives and USB devices with a virtual machine. It can simulate disk drives; an ISO image file can be mounted as:

- A virtual optical disc drive, and virtual hard disk drives are implemented as .vmdk files.
- VMware Workstation Pro can save the state of a virtual machine (a "snapshot") at any instant. These snapshots can later be restored, effectively returning the virtual machine to the saved state as it was and free from any post-snapshot damage to the VM.

UBUNTU:

Ubuntu is a complete Linux operating system, freely available with both community and professional support. The Ubuntu community is built on the ideas enshrined in the Ubuntu Manifesto: that software should be available free of charge, that software tools should be usable by people in their local language and despite any disabilities, and that people should have the freedom to customize and alter their software in whatever way they see fit.

- Ubuntu will always be free of charge, and there is no extra fee for the “enterprise edition”, we make our very best work available to everyone on the same Free terms.
- Ubuntu includes the very best in translations and accessibility infrastructure that the Free Software community has to offer, to make Ubuntu usable by as many people as possible.
- Ubuntu is shipped in stable and regular release cycles; a new release will be shipped every six months. Every two even years an Ubuntu long term support (LTS) release will become available, that is supported for 5 years. The Ubuntu releases in between (known as development or non-LTS releases) are supported for 9 month each.
- Ubuntu is entirely committed to the principles of open source software development; we encourage people to use open source software, improve it and pass it on.

Ubuntu is suitable for both desktop and server use. The current Ubuntu release supports Intel x86 (IBM-compatible PC), AMD64 (x86-64), ARMv7, ARMv8 (ARM64), IBM POWER8/POWER9 (ppc64el), IBM ZzEC12/zEC13/z14 and IBM

Ubuntu includes thousands of pieces of software, starting with the Linux kernel version 4.15 and GNOME 3.28, and covering every standard desktop application from word processing and spreadsheet applications to internet access applications, web server software, email software, programming languages.

CHAPTER 8

ALGORITHM DESCRIPTION

LSB Algorithm:

- LSB means least significant bit.
- The most common and popular method of modern day steganography is to make use of LSB of picture's pixel information.
- This technique works best when the file is longer than the message file and if image is grey scale.
- When applying LSB techniques to each byte of a 24 bit image ,three bits can be encoded into each pixel.

Message Embedding Procedure :

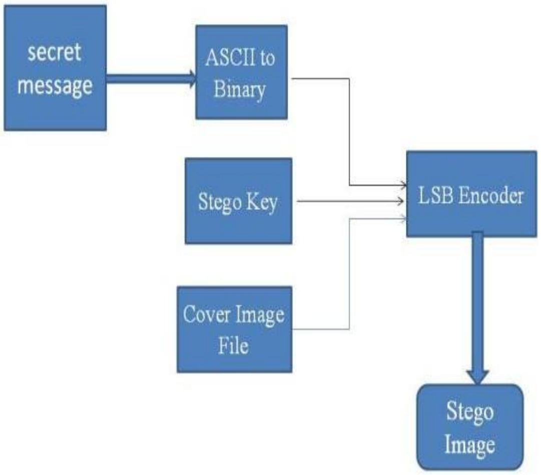
If the LSB of the pixel value of cover image C(i , j) is equal to the message bit SM of secret massage to be embedded, C(i , j) remain unchanged; if not, set the LSB of C(i , j) to SM.

- S(i, j) = C(i , j) - 1, if LSB(C(i , j)) = 1 and SM =0
- S(i, j) = C(i, j) + 1, if LSB(C(i , j)) = 0 and SM =1
- S(I, j) = C(i, j), if LSB(C(I , j)) =SM
- Where LSB(C(I , j)) stands for the LSB of cover image C(i , j) and "SM" is the next message bit to be embedded. S(I , j) is the stego image
- For example, suppose one can hide a message in three pixels of an image (24- bit colors). Suppose the original 3 pixelsare:[16]
- (11101010 1110100011001011)
- (01100110 11001010 11101000)
- (11001001 0010010111101001)

- A steganographic program could hide the letter "J" which has a position 74 into ASCII character set and have a binary representation "01001010", by altering the channel bits ofpixels.
- (11101010 111010011100101)
- (01100110 1100101111101000)

The following figure 1.11,1.12 shows the mechanism of LSB technique:

Fig: LSB Insertion Mechanism



Data Embedding:

The embedding process is as follows. Inputs Cover image, stego-key and the text Output stegoimage

Procedure :

Step 1: Extract the pixels of the cover image.

Step 2: Extract the characters of the text.

Step 3: Extract the characters from the Stego key.

Step 4: Choose 1st pixel and pick characters of the Stego key and place it in rst component of pixel.

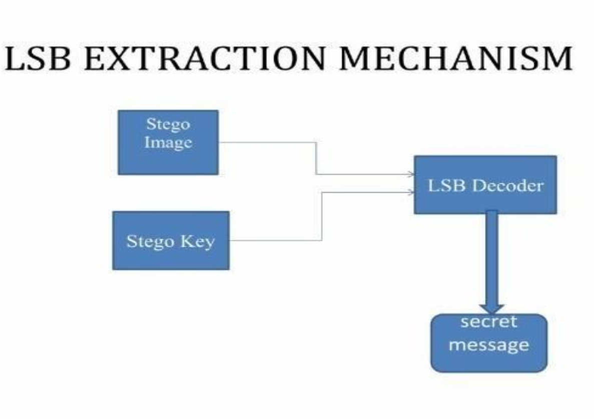
Step 5: Place some terminating symbol to indicate end of the key. 0 has been used as a terminating symbol in this algorithm.

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

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

Step 8: Again place some terminating symbol to indicate end of data. Step 9: Obtained stego image.

Fig: 1.12 LSB Extraction mechanism



Data Extraction:

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

Step 1: Extract the pixels of the stego image.

Step 2: Now, start from 1st pixel and extract stego key characters from 1st component of the pixels. Follow Step3 up to terminating symbol, otherwise follow step4..

Step 4: If this extracted key matches with the key entered by the receiver, then follow Step 5, otherwise terminate the program.

Step 5: If the key is correct, then go to next pixels and extract secret message characters from rst component of next pixels. Follow Step 5 till up to terminating symbol, otherwise follow step6.

Step 6: Extract secret message.

Pseudo-Random Encoding Technique:

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

Embedding Algorithm:

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

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

- Read character from text le that is to be hidden and convert the ASCII value ofthecharacterintoequivalentbinaryvalueintoan8bitintegerarray.
- Read the RGB colour image(cover image) into which the message is to be embedded.
- Read the last bit of red pixel.
- Initialize the random key and Randomly permute the pixels of cover image and reshape into a matrix.

- Initialize the stego-key and XOR with text to be hidden and give message.
- Insert the bits of the secret message to the LSB of the Redplane’s pixels.
- Write the above pixel to Stego ImageFile

Extraction of Hidden Message:

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

Message extraction algorithm:

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

1. Open the Stego image file in read mode and from the Image,
2. read the RGB colour of each pixel.
3. Extract the red component of the host image.
4. Read the last bit of each pixel.
5. Initialize the random-key that gives the position of the message bits in the red pixel that are embedded randomnly.
6. For decoding, select the pixels and Extract the LSB value of red pixels.
7. Read each of pixels then content of the array converts into decimal value that

is actually ASCII value of hidden character.

8. ASCII values got from above is XOR with stego-key and gives message file, which we hide inside the cover image.

CHAPTER 9

SYSTEM DESIGN

9.1 DATA FLOW DIAGRAM :

The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.

The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.

DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.

DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

Level 0:

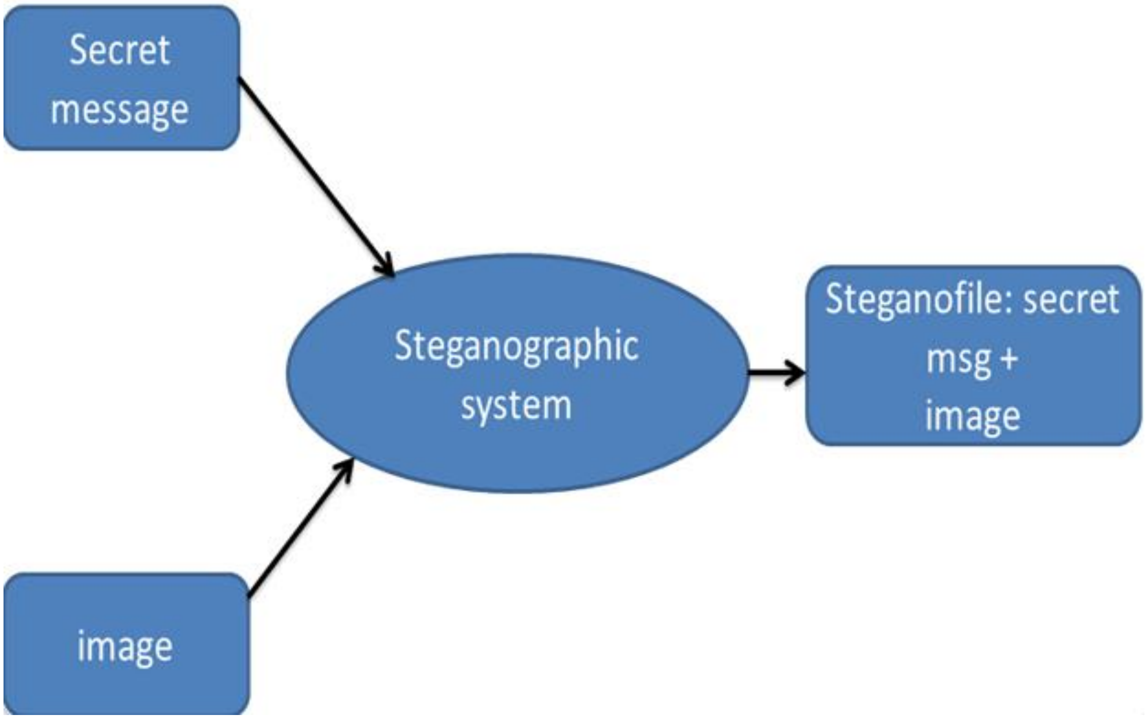


Fig : Data flow diagaram for image steganography

9.2 UML DIAGRAMS:

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

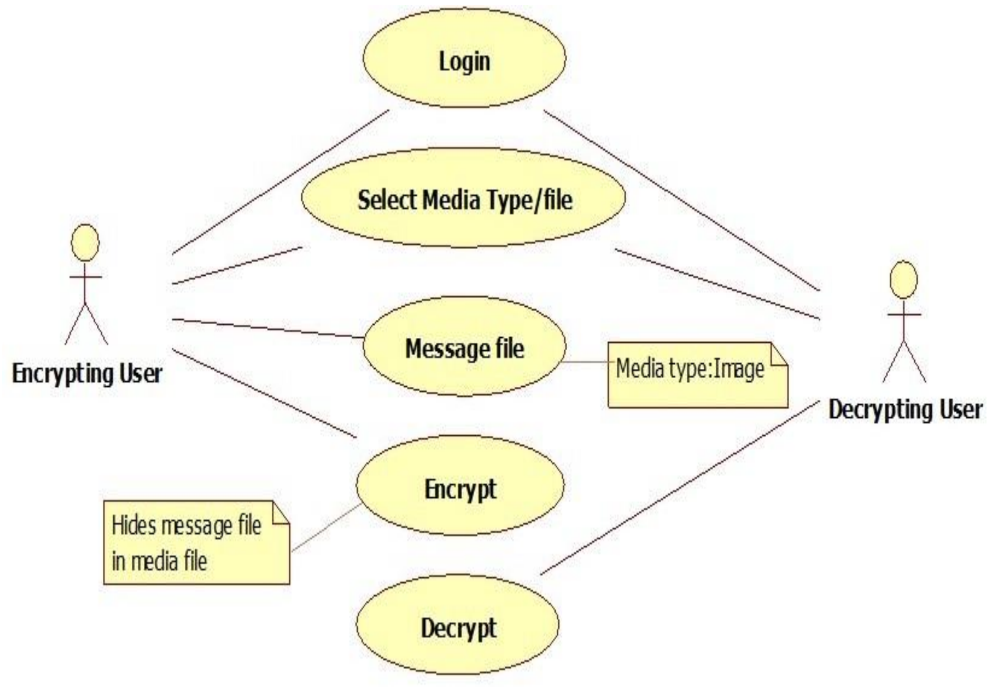
GOALS:

The Primary goals in the design of the UML are as follows:

- Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
- Provide extendibility and specialization mechanisms to extend the core concepts.
- Be independent of particular programming languages and development process
- Provide a formal basis for understanding the modeling language.
- Encourage the growth of OO tools market.
- Support higher level development concepts such as collaborations, frameworks, patterns and components.
- Integrate best practices

9.2.1 USE CASE DIAGRAM :

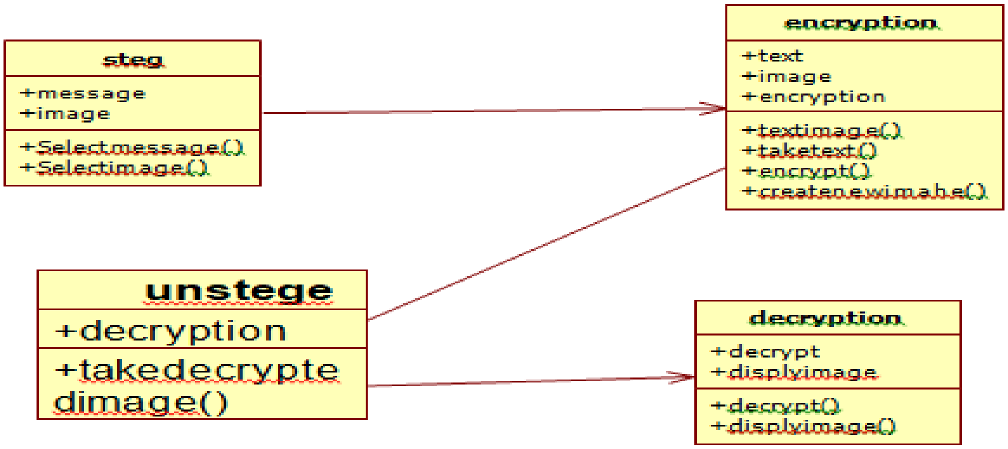
A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



Use case diagram for image steganography

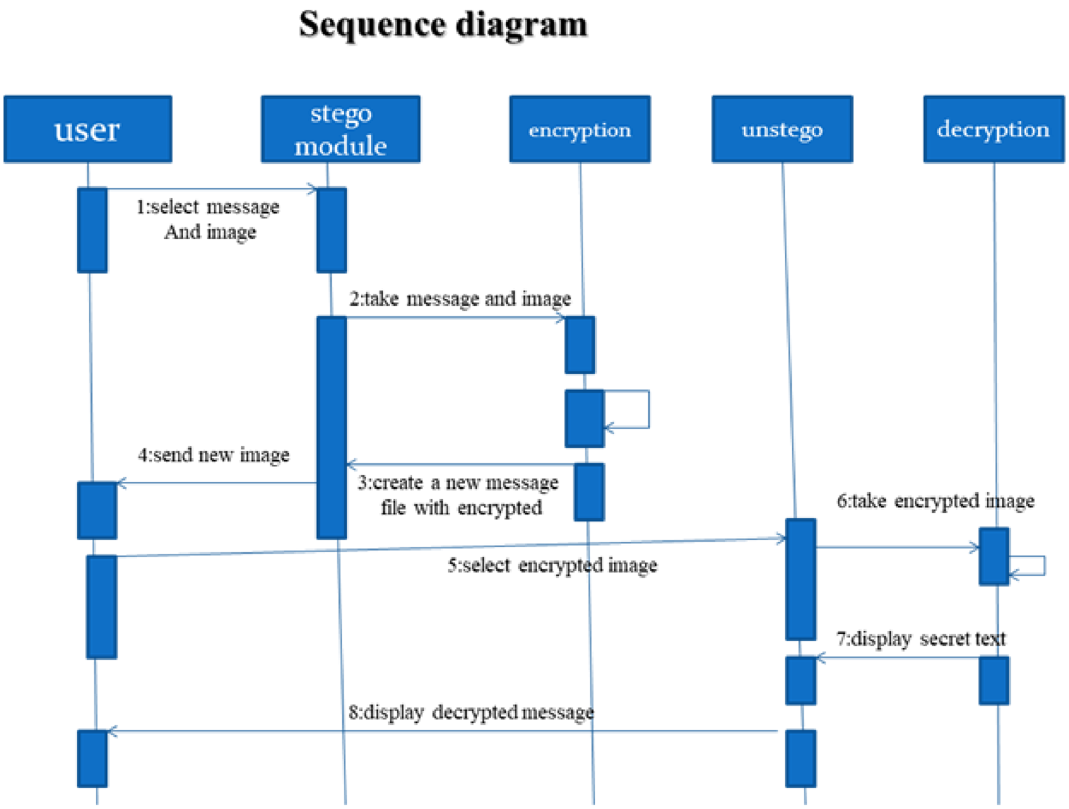
9.3. CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



9.4 SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



Sequence diagram for image steganography

CHAPTER 10

SAMPLE CODE

SAMPLECODE:

```
 Def put_binary_value(self, bits): #Put the bits In  the  image for c in bits:
        Val=list(self.image[self.curheight,self.curwidth])
         if int(c) == 1:
            val[self.curchan] = int(val[self.curchan]) | else: self.maskONE
            val[self.curchan] = int(val[self.curchan])
        &self.maskZEROself.image[self.curheight,self.curwidth] =
        tuple(val) self.next_slot()
def next_slot(self)
    ifself.curchan ==
         self.nbchannels-1:
         self.curchan = 0
        ifself.curwidth == self.width-
            1: self.curwidth = 0
            ifself.curheight == self.height-1:
                self.curheight = 0
                ifself.maskONE == 128:
                else:
                    self.maskONE = self.maskONEValues.pop(0)
                    self.maskZER=self.maskZEROValues.pop(0)
            else:
                self.curheight +=1
        else:
            self.curwidth +=1
    else:
        self.curchan +=1
```

CHAPTER 11

SCREENSHOTS

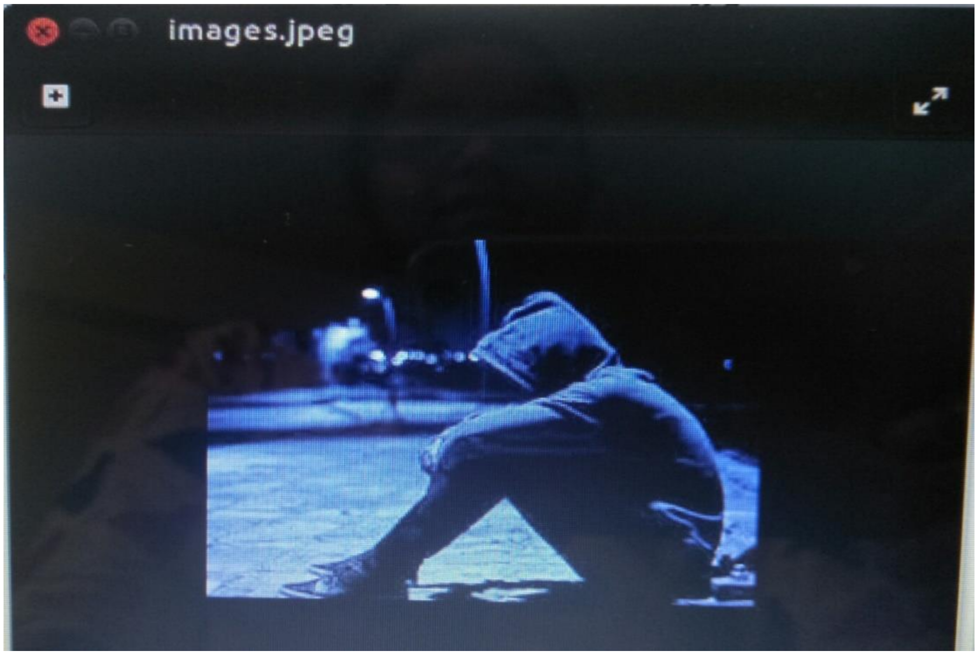
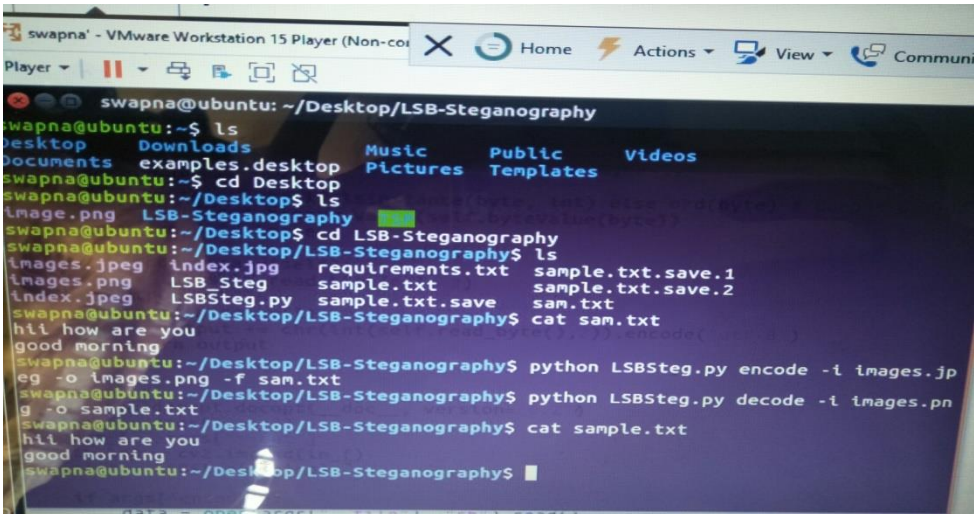


Fig: images.jpeg


swapna' - VMware Workstation 15 Player (Non-co
Home
Actions
View
Communi
Player
swapna@ubuntu: ~/Desktop/LSB-Steganography

```
wapna@ubuntu:~$ ls
esktop    Downloads          Music     Public     Videos
ocuments  examples.desktop   Pictures  Templates
swapna@ubuntu:~$ cd Desktop
swapna@ubuntu:~/Desktop$ ls
image.png  LSB-Steganography
swapna@ubuntu:~/Desktop$ cd LSB-Steganography
swapna@ubuntu:~/Desktop/LSB-Steganography$ ls
images.jpeg  index.jpg   requirements.txt  sample.txt.save.1
images.png   LSB_Steg    sample.txt        sample.txt.save.2
index.jpeg   LSBSteg.py  sample.txt.save   sam.txt
swapna@ubuntu:~/Desktop/LSB-Steganography$ cat sam.txt
hii how are you
good morning
swapna@ubuntu:~/Desktop/LSB-Steganography$ python LSBSteg.py encode -i images.jp
eg -o images.png -f sam.txt
swapna@ubuntu:~/Desktop/LSB-Steganography$ python LSBSteg.py decode -i images.pn
g -o sample.txt
swapna@ubuntu:~/Desktop/LSB-Steganography$ cat sample.txt
hii how are you
good morning
swapna@ubuntu:~/Desktop/LSB-Steganography$ 
```

Fig: Output

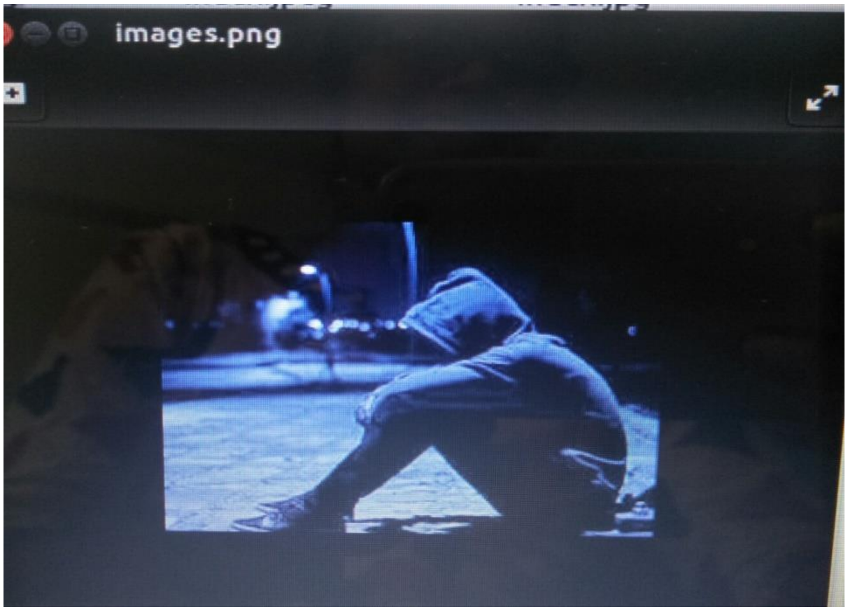


Fig: images.png

Fig Cover Image

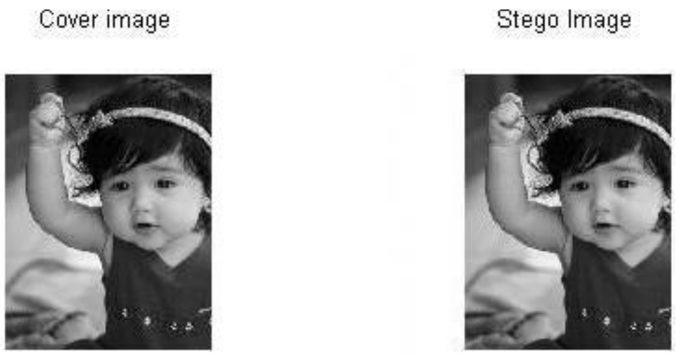


Figure: LSB technique

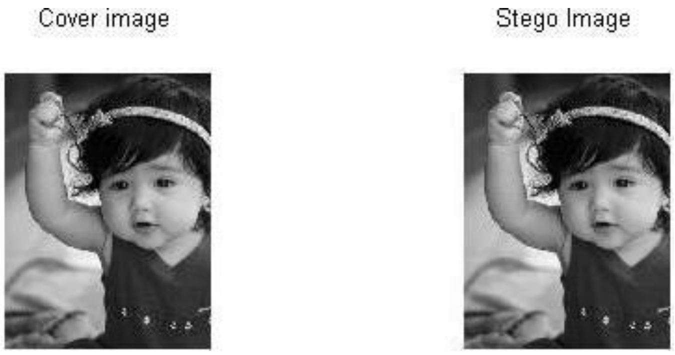


Figure: Pseudo-Random Encoding

Figure: RBG (cover image)

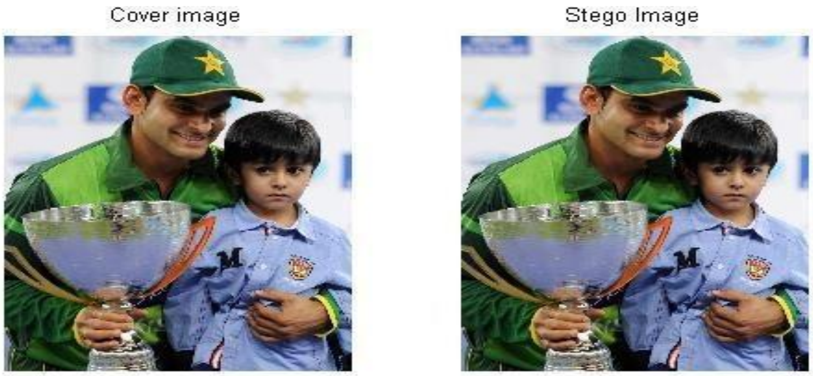
Figure: LSB Technique

Figure: Pseudo Random Technique

Figure: RBG Image

Fig: Secret Image

Fig 8.3.2: LSB Technique



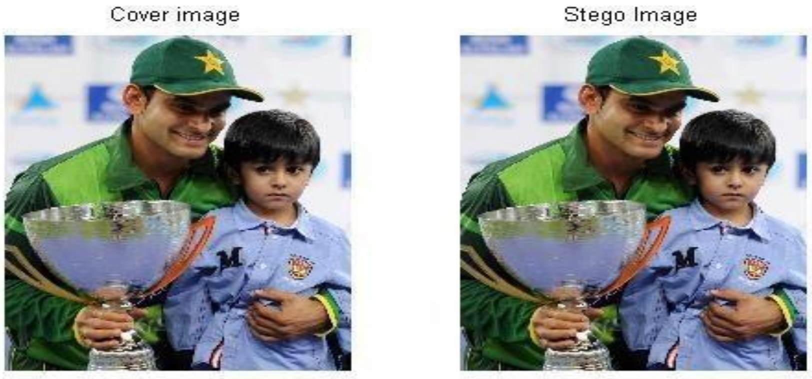
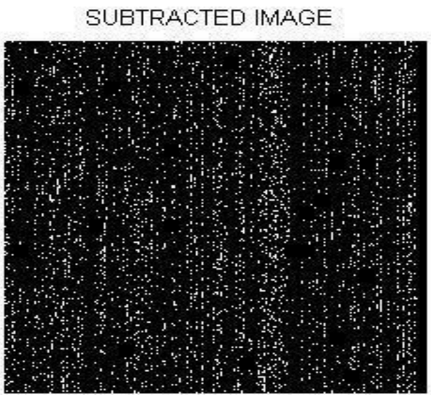


Figure: Pseudo random Technique

Figure: Dierence Image



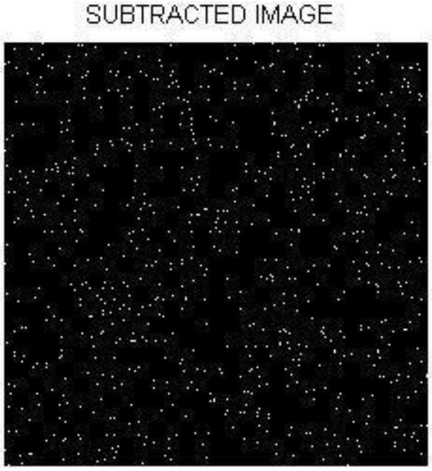


Figure: Difference Image

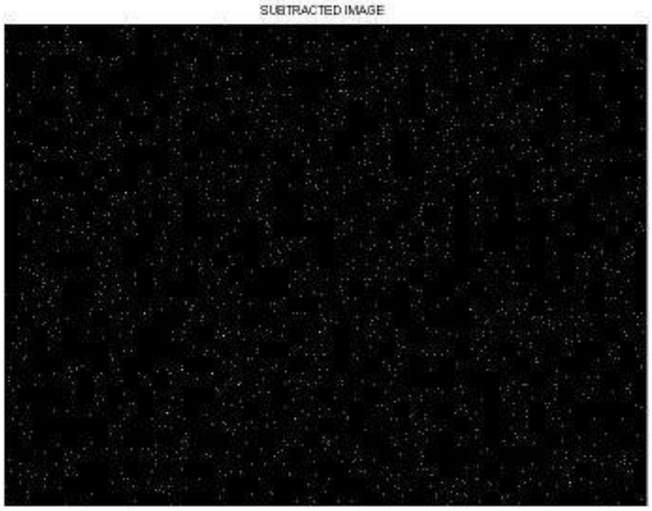
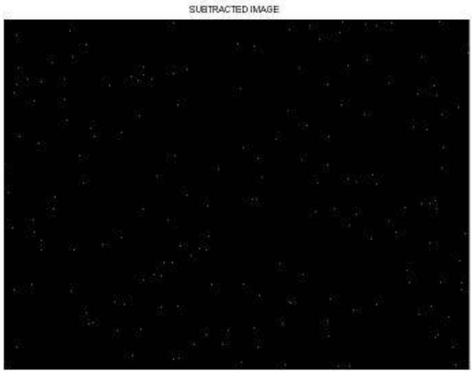


Figure:Difference image



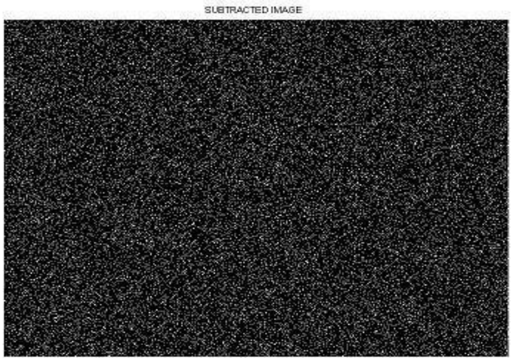


Figure:Difference image

Figure:Difference Image

CHAPTER 12

SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

12.1 TYPES OF TESTS:

12.1.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

12.1.2 Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully

unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

12.1.3 Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows;

data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

12.2 System Test:

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points

12.2.1 White Box Testing:

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

12.2.2 Black Box Testing:

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

CHAPTER 13

APPLICATION

- Secret Communications: The use steganography does not advertise secret communication and therefore avoids scrutiny of the sender, message, and recipient. A trade secret, blueprint, or other sensitive information can be transmitted without alerting potential attackers.
- Feature Tagging Elements can be embedded inside an image, such as the names of individuals in a photo or locations in a map. Copying the stegoimage also copies all of the embedded features and only parties who possess the decoding stegokey will be able to extract and view the features.
- Copyright Protection Copy protection mechanisms that prevent data, usually digital data, from being copied. The insertion and analysis of watermarks to protect copyrighted material is responsible for the recent rise of interest in digital steganography and data embedding.

CHAPTER 14

CONCLUSION

To hide confidential information steganography can be effectively used. The objective of any Steganographic method is to hide maximum secret information which is immune to external attacks and also should not convey the fact that the cover medium is carry secret information. This thesis has used LSB substitution technique for time domain and different transforms for frequency domain.

The random key made use of while LSB technique is employed has proved to be better than simple LSB substitution. When the personal is key is made use of in the techniques have not altered the resolution of the image much and appears to be negligible as been seen with the obtained results. Hence the hidden data getting damaged buy the third person is almost impossible. The algorithm can be implemented in both grayscale and color image since it has made use of 8 bit and 24bit images of size for both cover and secret image.

In Spatial domain methods one can get high payload capacity. The edge detection techniques which are used in the current methods do not recognizes the shades of the edge region, which can also be considered to embed the extra bits. The number of edge pixels can be increased by identifying the edges and shades of edges.

The Transform domain methods are highly robust. They embed the message bits in the regions which are highly insensitive to compression, filtration or transformation. But their payload capacity is low. Also the visual qualities of the Stego-imageare poor. Spatial domain is better compared to transform domain.In Spatial Domain techniques simple LSB substitution methods are less secure and payload capacity is less. To increase security Randomization techniques are used. These methods spread the message randomly into the cover image but payload capacity is still less. Edge detection method increases the payload capacity and it improves randomness and hence security. As discussed the current edge detection methods do not recognizes the shades of the edge region which are also capable of embedding more bits with less distortion. The shades ofthe edges are detected by Multiple Edge Detection method. In Multiple Edge Detection method the edge detection method is applied for 2-3 times which increase the number of edge pixels. As the number of edge pixels is increased more data can be hidden in to the cover image. To add randomness to the embedding procedure which increases security, Variable Embedding can be employed in which a suitable embedding ratio is chosen and according to that the message bits are embedded into the image pixels. The increase in the payload decrease PSNR value, which indicates image degradation amount. To improve PSNR Minimum Error Replacement [MER] method is used. In this method the next higher bit than the last embedded bit is toggle to decrease the pixelerror.

Stego and crypto way shows new way of embedding the data, especially in Multiresolution analysis, there are different ways of getting Multiresolution; this thesis has made use of Multiresolution analysis on wavelets and Curvelets. The experimental analysis has proved that Curvelets is the best Multiresolution transformation available. This work has been implemented with the library which has an accuracy of 15 fractional digits. The results obtained have a good PSNR value, along with the crypto style of embedding.

Steganography is to create secrete communication, in addition to this crypto way of embedding gives us higher end of security. Even if the person gets both stego and cove image he needs key to retrieve the data, without the key one can’t recover the data. Thus additional security is incorporated to the normal Steganography.

Image steganography is successfully obtained for different embedding techniques in both wavelet and curvelet domain. Depending on the demand of the application one can choose any of the techniques developed. The simulations are performed in MatLab 7.7.0(R2008b) and PSNR are calculated. It is found that Curvelet is a better approach than wavelet transformation. In this work data embedding is done by considering block of size85*85.

In the frequency domain technique use of skin tone algorithm has given very good results. The proposed method uses skin tone detection for finding skin portion of image and within this skin portion secret data embedding is done using DWT domain. Four cases of embedding are considered. It is observed that hiding of secret data in only the cropped skin portion enhances the security. And according to result and discussion proposed scheme provides good image quality in all four cases.

In Transform domain methods the pay load capacity is low and the future scope is to enhance Stego image quality and payload capacity. The payload capacity can still be increased by embedding more bits into edge pixels and less to non edge pixels. Future Enhancement can be done by embedding data pixel by pixel, thus increase in the payload can be attained.

CHAPTER 15

FUTURE WORK

In this work it explores only a small part of the science of steganography. As a new discipline, there is a great deal more research and development to do,

The following section describe areas for research which were offshoots of, or tangential to,our main objectives.

Detecting Steganography in Image

Can steganography be detected in images files? This is difficult question. It may be possible to detect a simple Steganographic technique by simple analyzing the low order bits of the image bytes. If the Steganographic algorithm is more complex, however, and spreads the embedded data over the image is random way or encrypts the data before embedding, it may be nearly impossible to detect.

How widespread is the Use of Steganography?

If a technique or set of techniques could be devised to detect steganography, it would be interesting to conduct a survey of images available on the internet to determine if steganography is used, by whom and for what purposes. Steganographic applications are available on the Internet, but it is not known if they are being used.

Steganography on the World Wide Web

The world wide web(www) makes extensive use of inline images. There are literally millions of images on various web pages worldwide. It may be possible to develop an application to serve as a web browser to retrieve data embedded in web page images. This “ stego-web” could operate on top of the existing WWW and be a means of covertly disseminating information.

Steganography in printed media.

If the data is embedded in an image, the image printed, then scanned and stored in a file can the embedded data be recovered? This would require a special form of a steganography to which could allow for in accuracies in the printing and scanning equipment.

Anti steganography measures

As was seen in this thesis, JPEG garbles any unencoded steganographically embedded data. Also, palettization (mapping a large number of colors in an image to a smaller subset of colors) of an image will it unsuitable for steganography. It is likely, as with JPEG, that some means may be employed to prevent loss of steganographically embedded data when its wrapper file is processed. The question remains open as to what is the most effective anti Steganographic tools or set of tools.

CHAPTER 16

REFERENCE

References Made From:

- User Interfaces in C#: Windows Forms and Custom Controls by Matthew MacDonald.
- Applied Microsoft® .NET Framework Programming (Pro-Developer) by Jeffrey Richter.
- Practical .Net2 and C#2: Harness the Platform, the Language, and the Framework by Patrick Smacchia.
- Data Communications and Networking, by Behrouz A Forouzan.
- Computer Networking: A Top-Down Approach, by James F.Kurose.
- Operating System Concepts, by Abraham Silberschatz.

Web Referred:

http://www.sourcefordgde.comhttp://www.networkcomputing.com/