Steganography

Peratti Technologies

PROJECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE OF MASTER OF SCIENCE
IN APPLIED SCIENCE - SOFTWARE ENGINEERING
OF BHARATHIAR UNIVERSITY, COIMBATORE.

Submitted by
Ms. S. Renuga Devi
Reg. no - 9937S0086

Under the Guidance of
Asst. Prof. K.R. Baskaran M.S.
Department of Computer Science and Engineering
Kumaraguru College Of Technology
Coimbatore.

Mr. L. Senthil Kumar
Project Leader
Peratti Technologies

Department of Computer Science and Engineering
Kumaraguru College of Technology
Coimbatore - 641 006.

MARCH 2004

PERATTI
TECHNOLOGIES

February 27,2004.

CERTIFICATE

This is to certify that the project report titled, "**STEGANOGRAPHY**" which is being submitted by Ms.S.Renuga Devi in partial fulfillment of the requirements for the award of the degree of Master of Science in Applied Science - Software Engineering of Bharathiar University, Coimbatore is a bonafide work carried out by her under the guidance of Mr. L.Senthil Kumar,Project Leader,at Peratti Technologies,Coimbatore during the period Nov 2003 to February 2004 to our satisfaction.

For Peratti Technologies,

Project Leader.
L.Senthil Kumar

CERTIFICATE

Department of Computer Science and Engineering
Kumaraguru College of Technology
Coimbatore - 641 006.

This is to certify that that the project work entitled
"Steganography"
has been submitted by

Ms. S. Renuga Devi

in partial fulfillment of the award of the degree of
Master of Science in Applied Science - Software Engineering of
Bharathiar University, Coimbatore.
During the academic year 2003-2004

Guide

Head of the Department

Certified that the candidate was examined by us in the Project Work Viva Voice Examination held on 29. 3. 2004 and the University Register Number was **9937S0086** .

29/3/04

Internal Examiner

External Examiner

DECLARATION

I here by declare that this project work entitled "Steganography" is a record of original project work done by me under the guidance of Asst. Prof. Mr. K.R. Baskaran, Department of Computer Science and Engineering as internal guide and Mr. L. Senthil Kumar, Project Leader as external guide, and this project work has not formed the basis for the award of any Degree / Diploma / Associate ship / Fellowship on similar titles to any other candidates of any university.

S. Renuga Devi

DATE: 12.3.2004

S. Renuga Devi

Internal Guide,
Asst. Prof. Mr. K.R. Baskaran,
Department of Computer Science and Engineering,
Kumaraguru College Of Technology,
Coimbatore.

External Guide,
Mr. L. Senthil Kumar,
Project Leader,
Peratti Technologies,
Coimbatore.

Dedicated to my beloved

Parents and Teachers

Acknowledgements

I would like to begin with a special note of gratitude and with my sincere and heart felt thanks to our Principal **Dr. K. K. Padmanaban B.Sc. (Engg), M.Tech, Ph.D** and our HOD **Prof. Dr. S. Thangaswamy Ph.D** for giving me the needed encouragement in starting this project and carrying out successfully.

I also take an immense pleasure especially in thanking our course coordinator and my internal project guide **Mr. K.R. Baskaran**, Asst.Professor, Dept of Computer Science and Engineering for constantly encouraging me to pursue new goals and ideas and have given his tremendous guidance and suggestions throughout my project.

I would like to express my sincere thanks to **Mr. K.K.S. Siva Prakash** , Managing Director of **Perrati Technologies**, for giving me an opportunity to do the project in their organization.

My grateful thanks to **Mr. L. Senthil Kumar** for rendering his excellent guidance for the successful and timely completion of the project.

I am proud of my parents and relatives for their unending support, love, understanding and encouragement throughout the endeavor.

I also thank my friends for encouraging me to finish this project in a successful way.

Above all, I thank God Almighty, who had always showered abundant blessings on me.

Synopsis

SYNOPSIS

The project entitled "Steganography", is being developed for "Perrati Technologies", a software development company, Coimbatore. The company deals in developing software for image processing and embedded systems.

Steganography, is an art of hiding information in a cover document or file. The goal of Steganography is to hide the messages in a way that it does not allow any enemy to even detect that there is a second secret message present.

The project mainly deals with **Identification of Image File Format**, **Hiding Process**, **Cryption** and **Retrieval Process**.

The images that are dealt in this project are BMP (8,16 and 24 bit image), GIF (8 bit image) and JPEG (24 bit image). As a first process of **identification of image file format** the details of the images are to be known so that we can know where to hide the data. Each and every image will have different file format specifications and that according to the specifications the details of the image will be retrieved and read. After the identification of image file format, we get the details of the image such as its file size, height, width, number of bits per pixel, resolution, the colors that are used and from which byte the raster or the pixel data starts. In this process we come to know the location of the raster data or pixel data, where we are going to embed the data that is to be hidden.

In the **hiding process** we make use of the Least Significant Bit insertion method. In this process we are embedding the hidden medium to the Least Significant Bit of the pixel data or raster data. Changes to the Most Significant Bit will result in drastic change in color and image quality. Whereas changes to the Least Significant Bit will have minimal impact such that the human eye cannot detect the changes.

Cryption is a science of secret writing, which involves encrypting and decrypting the information. Before hiding, the message can be encrypted and then hidden so that it maintains effective secrecy. Here cryption process is done by EX-OR method. In this method the entire process will be based on the secret keys given by the user.

In the **retrieval process**, the data is to be retrieved from the Least Significant Bit of the cover medium, that is within, which file the data is hidden. After retrieving if the data is found to be encrypted then it has to under go the process of decryption to get the original message.

INDEX

Topic Page No.

1. INTRODUCTION

1.1 ABOUT STEGANOGRAPHY:

Steganography is an art of hiding information in a cover document or file. Goal of steganography is to hide the messages in a way that it does not allow any enemy to even detect that there is a second secret message present. Steganography is used on text, images, sound and signals.

Steganography can be applied for reducing the network traffic, and thus reducing the consumption of the network bandwidth. It is used in hiding the private information and it even hides the existence of the information within the other medium.

When working on internet, it takes too much of our time to download or upload files. In particular, image files take much more time to transfer because of their larger size. These large files occupy most of our allocated network bandwidth. In order to make the file transfer very simple, we can transfer the two files at the same time. This reduces the network traffic and thus the consumption of network bandwidth.

Thus it is applied in bandwidth-reduced file transfer technique, which illustrates that suppose we have to transfer one image file and one text file from machine 'A' to machine 'B'. Let us assume that the image file takes 6 minutes to transfer and the text file takes 3 minutes to transfer from machine 'A' to machine 'B'. If we transfer image and the text files separately, it will take a total of 9 minutes to transfer. By using the bandwidth-reduced file transfer technique by applying Steganography we can transfer both the text file and the image file in 6 minutes. This way we save 3 minutes of our time, with reduced network traffic.

Secondly, in addition to the bandwidth-reduced file transfer technique, Steganography is also applied to hide the secure information and for the secure transfer and retrieval of information. In this process of hiding the secure information in order to provide more security to the information we also make use of cryptography.

Applications are as follows:

- Reduces network traffic and thus the consumption of network bandwidth.
- Used in hiding the private information and it even hides the existence of the information within the other medium.
- Used to combine explanatory information with an image (like doctor's notes accompanying an X-ray)

1.2 ORGANIZATION PROFILE:

Perrati Technologies is a leading hardware and software development company situated in Coimbatore. Here high quality standards are maintained to meet the requirements of world leaders.

Mr. K.K.S. SivaPrakash, Managing Director of Perrati Technologies, is the visionary and force behind Perrati Technologies. He takes care of long term planning and strategies. He has expertise in Finance, Costing, Software, Hardware, Project Consultancy and Management.

Mr. L. Senthil Kumar , Head - Software Development has a strong background in Software Engineering and has over 9 years of professional experience in Design, Development and Implementation of software projects. He has developed and implemented projects for large international clients.

Mission:

Our Greed is to deliver on time and all the while adding value to every project undertaken. Peratti Technologies is a consortium of experience, to analyze, innovate, and formulate solutions on various technologies and upgrading the skill levels.

1.3 ABOUT THE PROJECT:

The project Steganography deals with Image file format identification, Cryption, Data Hiding in BMP image, Data Retrieval from BMP Image, Data Hiding in GIF image, Data Retrieval from GIF Image, Data Hiding in JPEG image, Data Retrieval from JPEG Image. The image file formats are to be identified according to the file format specifications of BMP, GIF and JPEG. After the image file format identification we get the details of the image such as its file size, height, width, number of bits per pixel, resolution, the colors that are used and from which byte the raster or the pixel data starts, in which we are going to hide the message.

1.3.1 EXISTING SYSTEM AND ITS LIMITATIONS:

At present the company doesn't have any software for Steganography. A new software is to be developed for this purpose so it can be included in the software section of Perrati Technologies.

1.3.2 PROPOSED SYSTEM AND ITS LIMITATIONS:

The proposed system deals with embedding an image file within an image file or embedding a text file within an image file. In this case the software will be designed such that the user selects the two files, an image and a text file, where the image is the cover media, in which the text file is to be embedded directly, or it will be embedded after encrypting the text file. On the other hand two image files can be embedded and transferred. Steganography concept by itself is secure as it does not even show the existence of the information within the other medium. As in the case to provide more security to the private information we combine the concept of Cryptography along with the Steganography. This concept is very much useful for transfer of secure information and it can also be used to combine explanatory information

wing an X-ray

The images that are handled are BMP (8,16 and 24 bit image),GIF (8 bit image),JPEG (24 bit image).

The software with some modifications can be applied to most of the different types of images.

The Steganography comprises of ten modules they are

- Image file format identification for BMP
- Image file format identification for GIF
- Image file format identification for JPEG
- Cryption
- Data Hiding in BMP image
- Data Retrieval from BMP Image
- Data Hiding in GIF image
- Data Retrieval from GIF Image
- Data Hiding in JPEG image
- Data Retrieval from JPEG Image.

BASIC PROCESSES INVOLVED IN STEGANOGRAPHY:

Basically in Steganography there involves three important processes. They are

- BCD Representation
- Hiding Process
- Retrieval Process

BCD Representation:

This BCD representation is the base for Cryptography and Steganography. In Cryptography we make use of these bits to EX-OR. In Steganography we use these bits in iding and retrieving the message. Therefore we are in need of bitwise representation of all the characters that are present in the cover medium and hidden medium.

The explanation of BCD representation is illustrated with an example.

For example, let us consider we need to store the bitwise representation of 'a'. The ASCII value of 'a' is 97.Then the hexadecimal value of 97 is 61. Now we consider 61, we have the BCD representation of 61 as per the 8421 code as 0110 0001. Now it is required that we have to store each and every bit separately so that these bits can be manipulated latter.

The following calculation shows how we store the bitwise representation of 'a'.

As we know the BCD representation of 'a' is 0110 0001, the following are the steps to be followed.

1. AND 0110 0001 with 1000 0000 which results in 0000 0000 here bit[0] = 0
2. Then right shift 0th bit of 1000 000 that results in 0100 0000
3. Now AND 0110 0001 with 0100 0000 which results in 0100 0000 here bit[1] = 1
4. Then right shift 1st bit of 0100 000 that results in 0010 0000
5. Now AND 0110 0001 with 0010 0000 which results in 0010 0000 here bit[2] = 1
6. Then right shift 2nd bit of 0010 000 that results in 0001 0000
7. Now AND 0110 0001 with 0001 0000 which results in 0000 0000 here bit[3] = 0
8. Then right shift 3rd bit of 0001 000 that results in 0000 1000
9. Now AND 0110 0001 with 0000 1000 which results in 0000 0000 here bit[4] = 0
10. Then right shift 4th bit of 0000 1000 that results in 0000 0100
11. Now AND 0110 0001 with 0000 0100 which results in 0000 0000 here bit[5] = 0
12. Then right shift 5th bit of 0000 0100 that results in 0000 0010
13. Now AND 0110 0001 with 0000 0010 which results in 0000 0000 here bit[6] = 0
14. Then right shift 6th bit of 0000 0010 that results in 0000 0001
15. Now AND 0110 0001 with 0000 0001 which results in 0000 0001 here bit[7] = 1

Thus we obtain

bit[0]=0 bit[1]=1 bit[2]=1 bit[3]=0 bit[4]=0 bit[5]=0 bit[6]=0 bit[7]=1

Combining these values we obtain the value 0110 0001 whose hexadecimal value is 61. Now decimal value of 61 is 97. now this 97 is the ASCII value of 'a' and that we obtain 'a'.

0110 0001 ⟶ 61 (HEX value) ⟶ 97 (DEC value) ⟶ 'a' (ASCII value of

The calculation is shown below for a clearer view.

```
Eg: a = 97(ASCII) → (HEX of 97 = 61) BCD of 61 →  0110 0001
                                                   1000 0000
                                                 -----------
                                                   0000 0000
                                                   0100 0000
                                                 -----------
                                                   0100 0000
                                                   0010 0000
                                                 -----------
                                                   0010 0000
                                                   0001 0000
                                                 -----------
                                                   0000 0000
                                                   0000 1000
                                                 -----------
                                                   0000 0000
                                                   0000 0100
                                                 -----------
                                                   0000 0000
                                                   0000 0010
                                                 -----------
                                                   0000 0000
                                                   0000 0001
                                                 -----------
                                                   0000 0001
                                                 -----------
```

0110 0001 ⟶ 61 (HEX value) ⟶ 97 (DEC value) ⟶ 'a' (ASCII value of 'a'=97)

Hiding Process (Lsb Insertion Method):

In the hiding process we make use of the Lsb insertion method. In this process we are embedding the hidden medium to the last bit of Lsb. For example, if a character ‘A‘ is to be stored within an image. Character ‘A’ is of one byte and that the 8 bits of the character ‘A’ is stored in the following 8 bytes of the image in the Lsb.

Pixels of the Image (Cover Medium)

(00100111 11101001 11001000)
(00100111 11001000 11101001)
(11001000 00100111 11101001)

Let the character to be hidden be ‘A’

A: 0100 0001

Compare the first bit of BCD value of ‘A’ with the last Lsb of 00100111, here the last Lsb is 1 which is to be converted to 0 to embed first bit of BCD value of ‘A’. Next compare the second bit of BCD value of A which is 1 with the last Lsb of 11101001 here the last Lsb is also 1 so there is no need of conversion. This process is to be done for all the bits of ‘A’.

So finally after comparing each and every bit of ‘A’ with that of Lsb of each and every byte of the image pixels, we obtain the result where the entire BCD value of A is embedded within the pixels.

Result: 0010011**0** 1110100**1** 1100100**0**
0010011**0** 1100100**0** 1110100**0**
1100100**0** 0010011**1**

Finally we come to the conclusion that for embedding 1 byte of hidden medium we need 8 bytes of cover medium.

Retrieval Process:

For example, if a character 'A' is stored within an image. Character 'A' is of one byte and that the 8 bits of the character 'A' is stored in the following 8 bytes of the image in the last bit of Lsb.

0010011**0** 1110100**1** 1100100**0**
0010011**0** 1100100**0** 1110100**0**
1100100**0** 0010011**1**

During the retrieval process only the last bit of Lsb is retrieved from all the 8 bytes of the image and that it results in 0100 0001.

Char	ASCII	HEX	BCD
A	65	41	0100 0001

The following calculation is done to the retrieved bits in order to get the original information.

0 1 0 0 0 0 0 1

←

(1*2 pow 0) + (0*2 pow 1)+ (0*2 pow 2) + (0*2 pow 3)+ (0*2 pow 4) + (0*2 pow 5) +
(1*2 pow 6) +(0*2 pow 7)
= 1+64 =>65 => which is ASCII value of 'A'.

The description of the modules are illustrated below.

1. Image file format identification for BMP:

The bitmap image file refers to the standard Windows image format. A bitmap file is a raster (or pixel) based format that only supports the RGB color space and bit depths of 1, 4, 8, or 24 bits per pixel. Even though bitmap images are in the RGB color space, they are not supported by any Web browsers or Web coding languages. Therefore, they are not suitable for use as images in a Web application. Bitmap images are best used for their intended purpose, as a system support on a PC Windows-based computer.

The following file format specification describes the details of the image, such that it is required to retrieve the number of bytes as given in the specification to obtain the information about the image.

Name	Size	Description
Header	14 bytes	Windows Structure: BITMAPFILEHEADER
Signature	2 bytes	'BM'
FileSize	4 bytes	File size in bytes
Reserved	4 bytes	Unused (=0)
DataOffset	4 bytes	File offset to Raster Data
InfoHeader	40 bytes	Windows Structure: BITMAPINFOHEADER
Size	4 bytes	Size of InfoHeader =40
Width	4 bytes	Bitmap Width
Height	4 bytes	Bitmap Height
Planes	2 bytes	Number of Planes (=1)
BitCount	2 bytes	Bits per Pixel 1 = monochrome palette. NumColors=1 4 = 4bit palletized. NumColors = 16 8 = 8bit palletized. NumColors = 256 16 = 16bit RGB. NumColors = 65536 24 = 24bit RGB. NumColors = 16777216
Compression	4 bytes	Type of Compression 0 = BI_RGB no compression 1 = BI_RLE8 8bit RLE encoding 2 = BI_RLE4 4bit RLE encoding
ImageSize	4 bytes	Compressed Size of Image It is valid to set this =0 if Compression = 0
XpixelsPerM	4 bytes	horizontal resolution: Pixels/meter
YpixelsPerM	4 bytes	vertical resolution: Pixels/meter
ColorsUsed	4 bytes	Number of actually used colors
ColorsImportant	4 bytes	Number of important colors 0 = all
ColorTable	4 * NumColors bytes	present only if Info.BitsPerPixel <= 8 colors should be ordered by importance
Red	1 byte	Red intensity

Green	1 byte	Green intensity
Blue	1 byte	Blue intensity
Reserved	1 byte	unused (=0)
Raster Data	Info.ImageSize bytes	The pixel data

Here the header contains 14 bytes which includes the Signature, File Size, Reserved and Data Offset.

HEADER:

The following gives the description of the header information.

Signature- Indicates what type of file it is whether BMP, GIF, JPEG etc., and its corresponding version.

File Size-Gives the file size of the image.

Reserved- It is the space that is been left for addition of information to the image in future.

Data Offset- Indicates in bytes from where the Raster or the Pixel data starts.

INFORMATION HEADER:

The following gives the description of the details under the information header.

Size- Indicates the size of the information header.

Width-Indicates the width of the Bitmap image

Height- Indicates the height of the Bitmap image.

Planes- An imaginary plane on which the picture is projected.

Bit Count-Indicates the number of bits in a pixel such as 8 bits in a pixel,24 bits in a pixel etc.

Compression- Indicates whether the image is compressed or not. If compressed it also indicates the type of algorithm used for compression.

Image Size-Gives the size of the compressed image, if compression is not done in the image then the image size is zero.

X Pixels Per M-Indicates the horizontal resolution of the image in pixels/meter.

Y Pixels Per M-Indicates the vertical resolution of the image in pixels/meter.

Colors Used-Gives the total number colors used in the image.

Colors Important-Gives the number of important colors in the image, if this field is zero the all the colors used in the image are important.

Color Table-A color table will be maintained for up to 8-bit images. This table consists of the colors sorted by decreasing order of importance.

Red- Gives the intensity of Red color.

Green-Gives the intensity of Green color

Blue- Gives the intensity of Blue Color.

Reserved- It is the space that is been left for addition of information to the image in future.

Raster Data-Indicates the total size of the pixel data in bytes.

2. Image file format identification for GIF:

The Graphics Interchange Format is one of the most popular file formats for Web graphics and for exchanging graphics files between computers. It is most commonly used for images composed of line drawings or blocks of a few distinct colors. The GIF format supports 8 bits of color information or less.

The following file format specification describes the details of the image, such that it is required to retrieve the number of bytes as given in the specification to obtain the information about the image.

Name	Size	Description
Signature	6 bytes	'GIF87a' or 'GIF89a'
GlobalDescriptor	7 bytes	global descriptor, always present
Width	2 bytes	width in pixels
Height	2 bytes	height in pixels
Flags	1 byte	global descriptor flags.
GlobalColorMap	bit 7	=1 if GlobalColorMap exists (should be true in almost all cases) =0 if default map is used, or if every image has a LocalColorMap
ColorResolutionBits	bits 6-4	+1 = significant bits per color in GlobalColorMap
Reserved	bit 3	=0
PixelBits	bits 2-0	+1 = ColorDepth, NumberOfGlobalColors := $2^{ColorDepth}$
BackgroundColor	1 byte	background color number (from GlobalColorMap or default map)
AspectRatio	1 byte	usually present
GlobalColorMap	NumberOfGlobalColors * 3	global color table, present only when GlobalDescriptor.Flags.GlobalColorMap = 1
Red	1 byte	red intensity of color
Green	1 byte	green intensity of color
Blue	1 byte	blue intensity of color
LocalDescriptor	10 bytes	local descriptor always present
Header	1 bytes	Always present
PosX	2 bytes	horizontal position of image
PosY	2 bytes	vertical position of image
Width	2 bytes	width of image
Height	2 bytes	height of image
Flags	1 byte	local descriptor Flags
LocalColorMap	bit 7	=1 if LocalColorMap exists =0 if GlobalColorMap is used

InterlacedImage	bit 6	=1 Interlaced =0 Non Interlaced
Sorted	bit 5	usually =0
Reserved	bit 4-3	=0
PixelBits	bit 2-0	+1 = ColorDepth, NumberOfLocalColors := $2^{ColorDepth}$
LocalColorMap	NumberOfLocalColors * 3	local color table, present only when LocalDescriptor.Flags.LocalColorMap = 1
Red	1 byte	red intensity of color
Green	1 byte	green intensity of color
Blue	1 byte	blue intensity of color
Raster Data Block	cannot be pre calculated	always present

The following gives the description of image file format of GIF.

Signature- Indicates what type of file it is whether BMP, GIF , JPEG etc., and its corresponding version such as 'GIF87a' or 'GIF89a'.

GLOBAL DESCRIPTIOR

Width - Gives the width of the GIF image.

Height – Gives the height of the GIF image.

Flags- Indicates the presence of global descriptor.

Global Color Map – Indicates whether the global color map is present or not. If this field's value is 1 then global color map is present else if 0 then local color map is present.

Color Resolution Bits – These bits governs the fine resolution of the color of the image.

Reserved- It is the space that is been left for addition of information to the image in future.

Pixel Bits - Indicates the number of bits in a pixel such as 8 bits in a pixel.

Background Color – Gives the background color of the image from Global Color Map or Local Color Map.

Aspect Ratio – Gives the ratio of length of scanning the line horizontally on the image, to the distance covered vertically on the image by all the scanning lines.

Red-Gives the intensity of Red color.

Green-Gives the intensity of Green color.

Blue-Gives the intensity of Blue Color.

LOCAL DESCRIPTIOR

Header – Gives the header information of image.

POSX – Gives the starting value of the horizontal position of the image.

POSY - Gives the starting value of the vertical position of the image.

Width - Gives the width of the GIF image.

Height – Gives the height of the GIF image.

Flags- Indicates the presence of local descriptor.

Local Color Map – Indicates whether the local color map is present or not. If this field's value is 1 then local color map is present else if 0 then global color map is present.

Interlaced Image – A method of displaying a GIF image where the raster lines are spaced apart to provide a way of visualizing the general content of an entire image before all of the data has been processed. If this field's value is 1 then it is an interlaced image, or if 0 then it is a non - interlaced image.

Sorted- Indicates whether the colors in the local color table is sorted or not. This sorting will be of decreasing order of importance. If this field's value is 0 then the colors in the local color table is not sorted, or if 0 the the colors are sorted.

Reserved- It is the space that is been left for addition of information to the image in future.

Pixel Bits - Indicates the number of bits in a pixel such as 8 bits in a pixel.

Red-Gives the intensity of Red color.

Green-Gives the intensity of Green color

Blue-Gives the intensity of Blue Color.

Raster Data Block – From here the raster data or the pixel data starts.

3. Image file format identification for JPEG:

JPEG is a popular file compression format which allows the storage of high quality images in relatively small files. JPEG file format is used to store "photo realistic" images and any art work that requires high color depth. It supports lossy compression where it reduces file sizes by eliminating redundant or unnecessary image data. Many digital cameras automatically save images usinxg the JPEG format. JPEG is the most commonly used image file format on the World Wide Web.

A JPEG file consists of the following parts:

1. A Start of Image (SOI)
2. An Application Marker (APP0 Marker)
    1. APP0 length
    2. Identifier
    3. Version
    4. Units for X & Y densities
    5. X density
    6. Y density
3. A Start of Frame (SOF0)

    1. Start of Frame length
    2. Precision (Bits per pixel per color component)
    3. Image height
    4. Image width
    5. Number of color components

4. Define Restart Interval Marker (DRI Marker)
   Length
   Restart Interval (RI)

5. Define Huffman Table (DHT)
   Huffman Table Marker Identifier
   Length
   Huffman Table Information (HT Info)

6. A Start of Scan (SOS)
    1. Marker Identifier
    2. Length
    3. Number of color components

7. Image Data immediately follows the SOS segment.

APP0 marker :

The JPEG File Interchange Format (JFIF) is a minimal file format which enables JPEG bit streams to be exchanged between a wide variety of platforms and applications.

The JFIF APP0 marker provides information which is missing from the JPEG stream: version number, X and Y pixel density (dots per inch or dots per cm), pixel aspect ratio (derived from X and Y pixel density).

APP0 (JFIF segment marker) marker:

Field	**Size**	**Description**
Marker Identifier	2 bytes	0xff, 0xe0 to identify APP0 marker
Length	2 bytes	It must be >= 16
File Identifier Mark	5 bytes	This identifies JFIF. (0x4a, 0x46, 0x49, 0x46, 0x00)
Major revision number	1 byte	Should be 1, otherwise error

Minor revision number	1 byte	Should be 0..2
Units for x/y densities	1 byte	0 – no units 1 = x/y-density are dots/inch 2 = x/y-density are dots/cm
X-density	2 bytes	It should be <> 0
Y-density	2 bytes	It should be <> 0

Marker Identifier- Gives the starting address of APPO marker.

Length- Gives the length of the APPO marker.

File Identifier Mark-Gives the starting address of JFIF (JPEG File Interchange Format).

Major revision number-Gives the major revision number.

Minor revision number-Gives the minor revision number.

Units for x/y densities-Specifies the units in mentioning the quantity of pixels. whether in dots/inch or dots/centimeters.

X density- Number of pixels horizontally.

Y density- Number of pixels vertically.

SOF0 (Start Of Frame 0) marker:

Field	**Size**	**Description**
Marker Identifier	2 bytes	0xff, 0xc0 to identify SOF0 marker
Length	2 bytes	This value equals to 8 + components*3 value
Data precision	1 byte	This is in bits/pixel
Image height	2 bytes	This must be > 0
Image Width	2 bytes	This must be > 0
Number of components	1 byte	Usually 1 = grey scaled, 3 = color YcbCr or YIQ 4 = color CMYK

Marker Identifier- Gives the starting address of SOFO marker.

Length- Gives the length of SOFO marker.

Data Precision- This gives the number of bits present in a pixel.

Image height- Gives the height of the image.

Image width-Gives the width of the image.

Number of components- Gives the number of components present in an image.

Components- In JPEG a picture is broken down into a number of components, when these components are put back together the end image can be viewed.

DRI (Define Restart Interval) marker:

Field	**Size**	**Description**
Marker Identifier	2 bytes	0xff, 0xdd identifies DRI marker
Length	2 bytes	It must be 4
Restart interval	2 bytes	This is in units of MCU blocks.

Marker Identifier- Gives the starting address of DRI marker.

Length- Length of DRI marker.

Restart interval- JPEG divides the component data into independent decodable segments called restart intervals.

DHT(Define Huffman Table) marker:

Field	**Size**	**Description**
Marker Identifier	2 bytes	0xff, 0xc4 to identify DHT marker
Length	2 bytes	This specify length of Huffman table
HT information	1 byte	Gives the number of HT present.

larker Identifier- Gives the starting address of DHT marker.

ength- Length of DHT marker.

IT information- Gives the number Huffman tables present.

SOS (Start Of Scan) marker:

Field	Size	Description
Marker Identifier	2 bytes	0xff, 0xda identify SOS marker
Length	2 bytes	This must be equal to 6+2*(number of components in scan).
Number of Components in scan	1 byte	Usually 1 or 3
Ignorable Bytes	3 bytes	We have to skip 3 bytes.

Marker Identifier- Gives the starting address of SOS marker.

Length- Gives the length of SOS marker.

Number of components in scan-Gives the number of components present during the scan.

Note: Image Data immediately follows the SOS segment.

4. Cryption:

Cryption is a science of secret writing which involves encrypting and decrypting the information. Encryption is a process of translating a message, called the Plaintext, into an encoded message, called the Ciphertext. Decryption is the reverse process of Encryption. Frequently, the same Cipher is used for both Encryption and Decryption. While Encryption creates a Ciphertext from a Plaintext, Decryption creates a Plaintext from a Ciphertext. Secret Key is the key that is used for both encryption and decryption.

In most of the cryption processes, for encryption and decryption the same algorithm will be used. In such a case if the third person comes to know about the algorithm, he will be able to retrieve the secret messages easily. To avoid such a situation, here in this project the cryption process is done by EX-OR method, where the entire process is based on the passwords given by the user, and that the number of passwords can be of any number based on the wish of user.

Description of Cryption process by EX-OR Method :

Let the word that is to be encrypted be FORD. The following illustrates the process.

The first character of word FORD, 'F' is taken and its corresponding ASCII value, HEX value and BCD value is shown. Then the second character 'O' is taken and its corresponding ASCII value, HEX value and BCD value is shown, and this is repeated for all the characters in the word FORD.

Next the secret keys are given, that is secret keys are the passwords given by the user and in this process it can be of any number of passwords. Their corresponding BCD representation is also illustrated below.

FORD

Char	ASCII	HEX	BCD
F	70	46	0100 0110
O	79	4F	0100 1111
R	82	52	0101 0010
D	68	44	0100 0100

Secret Keys			
55	⟶	37	0011 0111
62	⟶	3E	0011 1110
72	⟶	48	0100 1000
[illegible]	⟶	50	0101 0000

Description of Encryption process by EX-OR Method :

In order to do encryption by EX-OR method, each and every character of the text file s to be read and we have to EX-OR each character with all of the passwords or secret keys. Finally the encoded message is obtained. The user has to give a file name as input so that this encoded message will be stored in that file name. Later this file containing the encoded message will be hidden within the cover medium.

As per the given example of FORD, we have to EX-OR the character ‘F’ with all of the secret keys given by the user, then the same procedure should be followed for the rest of the characters such as ‘O’,’R’ and ‘D’.

The following calculation illustrates the process of ‘F’ being encrypted. Here ‘R’ indicates the result of two values being EX-OR’ ed.

F	→	0100 0110	
55	→	0011 0111	(EX_OR)
R	→	0111 0001	
62	→	0011 1110	(EX_OR)
R	→	0100 1111	
72	→	0100 1000	(EX_OR)
R	→	0000 0111	
80	→	0101 0000	(EX_OR)
FR	→	0101 0111	(Encrypted Value Of ‘F’)

Similarly the rest of the characters in the word FORD has to under go the encryption process by EX-OR method.

Description of Decryption process by EX-OR Method :

During the process of encryption the encoded message has been stored in file. In the decryption process the encoded characters from the encoded file are read and each and every character is to be decrypted according to the passwords or secret keys given by the user. During decryption the user has to enter the secret keys correctly, as such entered during the encryption process. In this EX-OR method during the decryption process the user needn't enter the secret keys in the same order, and it can be of any order.

As per the given example of FORD, we have to EX-OR the character 'F' with all of the secret keys given by the user, then the same procedure should be followed for the rest of the characters such as 'O','R' and 'D'. The following calculation illustrates the process of 'F' being decrypted.

FR → 0101 0111 (Encrypted Value Of 'F')

80 → 0101 0000 (EX-OR)

R → 0000 0111

62 → 0011 1110 (EX-OR)

R → 0011 1001

72 → 0100 1000 (EX-OR)

R → 0111 0001

55 → 0011 0111 (EX-OR)

FR → 0100 0110

0100 0110 → Decimal 70 → F (original character)

Similarly the rest of the characters in the word FORD has to under go the decryption process by EX-OR method.

5. Data Hiding in BMP image:

The Data Hiding is done using the Lsb insertion method. For instance in the case of 8 bit image, each pixel will have 8 bits. The 4 bits to the left are the Most Significant Bits (Msb) and the 4 bits to the right are the Least Significant Bits (Lsb). If some changes are made to the Msb, it will result in drastic change in color and image quality. Hence we use the Lsb to hide the desired information. The human eye cannot detect the changes to only one or two bits in the Lsb.

For instance, if a bit pattern of 11001101 is changed to 11001100 they will essentially look the same to the naked eye.

The details of the given BMP image by the user is found from the Image File Format Identification for BMP module. From those details we know the starting of the pixel data or the raster data. Then we have to find out the size of the raster data. This size of the raster data should be eight times that of the text file or the image file that is to be hidden. If the size of the cover medium is correct to embed the hidden medium then the text or the image that is to hidden will be embedded within the pixel data using the Lsb insertion method.

6. Data Retrieval from BMP image:

As the user enters the image file or the cover medium within which the data is hidden in the form of text or image, the details of the cover medium are found out. From those details we know from where the raster data starts, within which the secret medium will be embedded. Then retrieval is done according to the retrieval process as described earlier.

7. Data Hiding in GIF Image:

The Data Hiding is done using the Lsb insertion method. For instance in the case of 8 bit image, each pixel will have 8 bits. The 4 bits to the left are the Most Significant Bits (Msb) and the 4 bits to the right are the Least Significant Bits (Lsb). If some changes are made to the Msb, it will result in drastic change in color and image quality. Hence we use the Lsb to ide the desired information. The human eye cannot detect the changes to only one or two its in the Lsb.

For instance, if a bit pattern of 11001101 is changed to 11001100 they will essentially look he same to the naked eye.

The details of the given GIF image by the user is found from the Image File Format Identification for GIF module. From those details we know the starting of the pixel data or the raster data. Then we have to find out the size of the raster data. This size of the raster data should be eight times that of the text file or the image file that is to be hidden. If the size of the cover medium is correct to embed the hidden medium then the text or the image that is to hidden will be embedded within the pixel data using the Lsb insertion method.

8. Data Retrieval from GIF image:

As the user enters the image file or the cover medium within which the data is hidden in the form of text or image, the details of the cover medium are found out, which gives the full details of the given cover medium. From those details we know from where the raster data starts, within which the secret medium will be embedded. Then retrieval is done according to the retrieval process as described earlier.

9. Data Hiding in JPEG Image:

The Data Hiding is done using the Lsb insertion method. For instance in the case of 8 bit image, each pixel will have 8 bits. The 4 bits to the left are the Most Significant Bits (Msb) and the 4 bits to the right are the Least Significant Bits (Lsb). If some changes are made to the Msb, it will result in drastic change in color and image quality. Hence we use the Lsb to hide the desired information. The human eye cannot detect the changes to only one or two bits in the Lsb.

For instance, if a bit pattern of 11001101 is changed to 11001100 they will essentially look the same to the naked eye.

The details of the given JPEG image by the user is found from the Image File Format Identification for JPEG module. From those details we know the starting of the pixel data or the raster data. Then we have to find out the size of the raster data. This size of the raster data should be eight times that of the text file or the image file that is to be hidden. If the size of the cover medium is correct to embed the hidden medium then the text or the image that is to hidden will be embedded within the pixel data using the Lsb insertion method.

10. Data Retrieval from JPEG image:

As the user enters the image file or the cover medium within which the data is hidden in the form of text or image, the details of the cover medium are found out, which gives the full details of the given cover medium. From those details we know from where the raster data starts, within which the secret medium will be embedded. Then retrieval is done according to the retrieval process as described earlier.

2. SYSTEM REQUIREMENTS

2.1 PURPOSE:

The project, “Steganography” is to be done in "Peratti Technologies", which is a software development company, that deals mainly in developing software for embedded systems. This software on Steganography is to be newly developed to be included in their software library.

2.2 PROBLEM STATEMENT:

The project involves two main concepts Steganography and Cryptography. Here Steganography, is an art of hiding information in a cover document or file. Goal of steganography is to hide the messages in a way that it does not allow any enemy to even detect that there is a second secret message present. Cryptography is combined with Steganography in order to provide more security to the hidden message.

The Steganography is to be done for Images and Text. Such that hiding an Image within another image and hiding the text within another image. In the second case that is while hiding the text within another image, in order to provide more privacy the text is encrypted and then hidden within the image. Then while retrieving the hidden text from the image, the text that was encrypted will be decrypted and that the original text will be obtained. Here the images that we are dealing with are BMP (8,16 and 24.bit image), GIF (8 bit image) and JPEG (24 bit image).

2.3 SCOPE:

Steganography is an art of hiding information in a cover document or file. That is, the goal of steganography is to hide the messages in a way that does not allow any enemy to even detect that there is a second secret message present. Steganography can be applied for reducing the etwork traffic, and thus reducing the consumption of the network bandwidth. Used in hiding the rivate information and it even hides the existence of the information within the other medium.

When working on internet, it takes too much of our time to download or upload files. In particular, image files take much more time to transfer because of their larger size. These large files occupy most of our allocated network bandwidth. In order to make the file transfer very simple, we can transfer the two files at the same time. This reduces the network traffic and thus the consumption of network bandwidth.

Thus it is applied in bandwidth-reduced file transfer technique, which illustrates that suppose we have to transfer one image file and one text file from machine 'A' to machine 'B'. Let us assume that the image file takes 6 minutes to transfer and the text file takes 3 minutes to transfer from machine 'A' to machine 'B'. If we transfer image and the text files separately, it will take a total of 9 minutes to transfer. By using the bandwidth-reduced file transfer technique by applying Steganography we can transfer both the text file and the image file within 6 minutes. This way we save 3 minutes of our time, with reduced network traffic.

Secondly, in addition to the bandwidth – reduced file transfer technique, Steganography is also applied to hide the secure information and for the secure transfer and retrieval of information. In this process of hiding the secure information in order to provide more security to the information we also make use of cryptography.

Applications are as follows:

- Reduces network traffic and thus the consumption of network bandwidth.
- Used in hiding the private information and it even hides the existence of the information within the other medium.
- Used to combine explanatory information with an image (like doctor's notes accompanying an X-ray)

.4 OVERVIEW:

The project "Steganography" deals with two main concepts Steganography and Cryptography. Here Steganography, is an art of hiding information in a cover document or file. Goal of Steganography is to hide the messages in a way that it does not allow any enemy to even detect that there is a second secret message present. Cryptography is combined with Steganography in order to provide more security to the hidden message.

The Steganography is to be done for Images and Text. Such that hiding an Image within another image and hiding the text within another image. In the second case that is while hiding the text within another image, in order to provide more privacy the text is encrypted and then hidden within the image. Then while retrieving the hidden text from the image, the text that was encrypted will be decrypted and that the original text will be obtained.

Here the images that we are dealing with are BMP (8,16 and 24 bit image), GIF (8 bit image) and JPEG (24 bit image).

It deals with Image file format identification of BMP, GIF and JPEG, Cryption, Data Hiding in BMP image, Data Retrieval from BMP Image, Data Hiding in GIF image, Data Retrieval from GIF Image, Data Hiding in JPEG image, Data Retrieval from JPEG Image. The image file formats are to be identified according to the file format specifications of BMP, GIF and JPEG. After the image file format identification we will get the details of the image such as its file size, height, width, number of bits per pixel, resolution, the colors that are used and from which byte the raster or the pixel data starts, where we are going to hide the message.

As the function of the product is to embed two files such that embedding an image file within an image file or embedding a text file within an image file. In this case the software will be designed such that the user selects the two files, an image and a text file, where the image is the cover media, in which the text file is to be embedded directly, or it will be embedded after encrypting the text file. On the other hand two image files can embedded and transferred. Steganography concept by itself is secure as it does not even show the existence of the information within the other medium. As in the case to provide more security to the private information we combine the concept of Cryptography along with the Steganography. In this crypting technique, encryption is done according to the users secret keys, which are to be got as input from the user and then the message is encrypted. The encrypted message will be hidden according to the concept of Steganography and then the entire Stego media, that is Carrier media with hidden message can be transmitted.

To retrieve the embedded file, the user selects the stego media and then when the user clicks the retrieve button the software asks for the password for authentication. This password should match with the password which was given by the user during the hiding process. After authentication is successful the embedded file is retrieved, if in the case if the embedded file is an encrypted file the user should provide the secret keys that where entered during the encryption process. Once secret keys are entered the decryption is done and that the original message is retrieved.

2.5 GENERAL DESCRIPTION

2.5.1 Product Perspective:

The software Steganography, is such that embedding an image file within an image file or embedding a text file within an image file. In this case the software will be designed such that the user selects the two files, an image and a text file, where the image is the cover media, in which the text file is to be embedded directly, or it will be embedded after encrypting the text file. On the other hand two image files can be embedded and transferred. Steganography concept by itself is secure as it does not even show the existence of the information within the other medium. As in the case to provide more security to the private information we combine the concept of Cryptography along with the Steganography. This concept is very much useful for transfer of secure information and it can also be used to combine explanatory information with an image like doctor's notes accompanying an X-ray.

Definitions:

- Steganography- Steganography is a means of storing the information in a way that hides private information and even the existence of the information within the other medium.
- Cover media- Cover media is the carrier medium, like text, image, audio, video.
- Secret message- Secret message is the private message that is to be hidden inside the Cover media.
- Stego media- Carrier with hidden message.
- Pixel- An abbreviation of the term 'picture element.' A pixel is the smallest picture element of a digital image.
- Resolution- The number of pixels in a graphic or screen in the horizontal and vertical dimensions, which governs the level of fine details the images can have.
- Cryptography-A system for encrypting and decrypting data is a cryptosystem.
- Encryption- Encryption is a process of translating a message, called the Plaintext, into an encoded message, called the Ciphertext.
- Decryption- Decryption is the reverse process to Encryption. Frequently, the same Cipher is used for both Encryption and Decryption. While Encryption creates a Ciphertext from a Plaintext, Decryption creates a Plaintext from a Ciphertext.
- Ciphertext- Ciphertext is encoded text, after it has been passed through an Encryption algorithm. It is the product of Plaintext after Encryption.
- Plaintext- Plaintext is an unencrypted message, before it is passed through an Encryption algorithm (Cipher). It is used to create a Ciphertext.
- Cipher- A Cipher is a computer software algorithm used for Encryption.
- Secret Key- The key that is used for both encryption and decryption
- Network Bandwidth- Refers to the data rate supported by a network connection or interface. One most commonly expresses bandwidth in terms of bytes per second (Bps).

2.5.2 *Product Functions:*

The product performs three main functions. They are

- Hiding the file
- Applying Cryption to the file
- Retrieving the file

Hiding the file:

In the case of hiding the file, two files are to be selected an image and a text file, where the image is the cover media, in which the text file is to be embedded directly, or it will be embedded after encrypting the text file. On the other hand two image files can be embedded and transferred. This concept is used to transfer files such as an image file followed by the text file, or where in the case of transferring two image files one by one at a time, which increases the network bandwidth and that in order to reduce the network bandwidth and the transfer time of the files we embed the files that are to be transferred and then we transfer the embedded file such that it reduces the network bandwidth and the transfer time. For instance it can be used to combine explanatory information with an image like doctor's notes accompanying an X-ray. This entire process is done such that as the user selects the cover media and the hidden message, a password is to be entered by the user so that this password authenticates for the retrieval process.

Applying Cryption to the file:

Steganography concept by itself is secure as it does not even show the existence of the information within the other medium. As in the case to provide more security to the private information we combine the concept of Cryptography along with the

Steganography. In this crypting technique, encryption is done according to the users secret keys, which are to be got as input from the user and then the message is encrypted.

The user has to give a file name as input so that this encoded message will be stored in that file name. Later this file containing the encoded message will be hidden within the cover medium according to the concept of Steganography.

In the case to decrypt an encrypted file the user should provide the secret keys that where entered during the encryption process. Once secret keys are entered the decryption is done and that the original message will be obtained.

Retrieving the file:

In the case to retrieve the embedded file, the user selects the stego media and then when the user clicks the retrieve button the software asks for the password for authentication. This password should match with the password, which was given by the user during the hiding process. After authentication is successful the embedded file is retrieved, if in the case if the embedded file is an encrypted file the user should provide the secret keys that where entered during the encryption process. Once secret keys are entered the decryption is done and that the original message is retrieved.

2.5.3 User Characteristics:

The person should have the knowledge of operating the computer. The software is designed as a user friendly product. The user can interact with the system by using the keyboard and mouse for giving the input. It is such that the user sets his own password during the hiding process and the user should use that password during the retrieval process.

.6 SPECIFIC REQUIREMENTS

2.6.1 Functional Requirements:

The various functions that are to be done are:

- Identifying the BMP image file format details to find out the location of raster or pixel data.
- Identifying the GIF image file format details to find out the location of raster or pixel data.
- Identifying the JPEG image file format details to find out the location of raster or pixel data.
- Applying encryption and decryption to the given input files.
- Applying the Hiding procedure for BMP image
- Applying the Data Retrieval procedure for BMP Image
- Applying the Hiding procedure for GIF image
- Applying the Data Retrieval procedure for GIF image
- Applying the Hiding procedure for JPEG image
- Applying the Data Retrieval procedure for JPEG image

2..6.2 Performance Requirements:

Security:

During hiding process the user has to enter the password, which is required for authentication later during the retrieval process. Along with this the secret keys that are to be entered by the user during the cryption process provides extra security for the private information.

Availability:

After giving the input, the process takes place and the output can be obtained within 1 minute.

Capacity:

For the entire process only two files are to be used either, one image and one text file or two image files. The size of the cover file should be 8 times that of the file that is to be hidden.

2.6.3 Design Constraints

Hardware Requirements:

Processor Pentium III 550 MHz

Floppy Disk Drive 1.44 MB

Hard Disk 20GB

System RAM 64MB

Software Requirements:

Operating System: Windows '95

Language : 'C'

User Interfaces:

The user can interface through keyboard and mouse in order to get the required details.

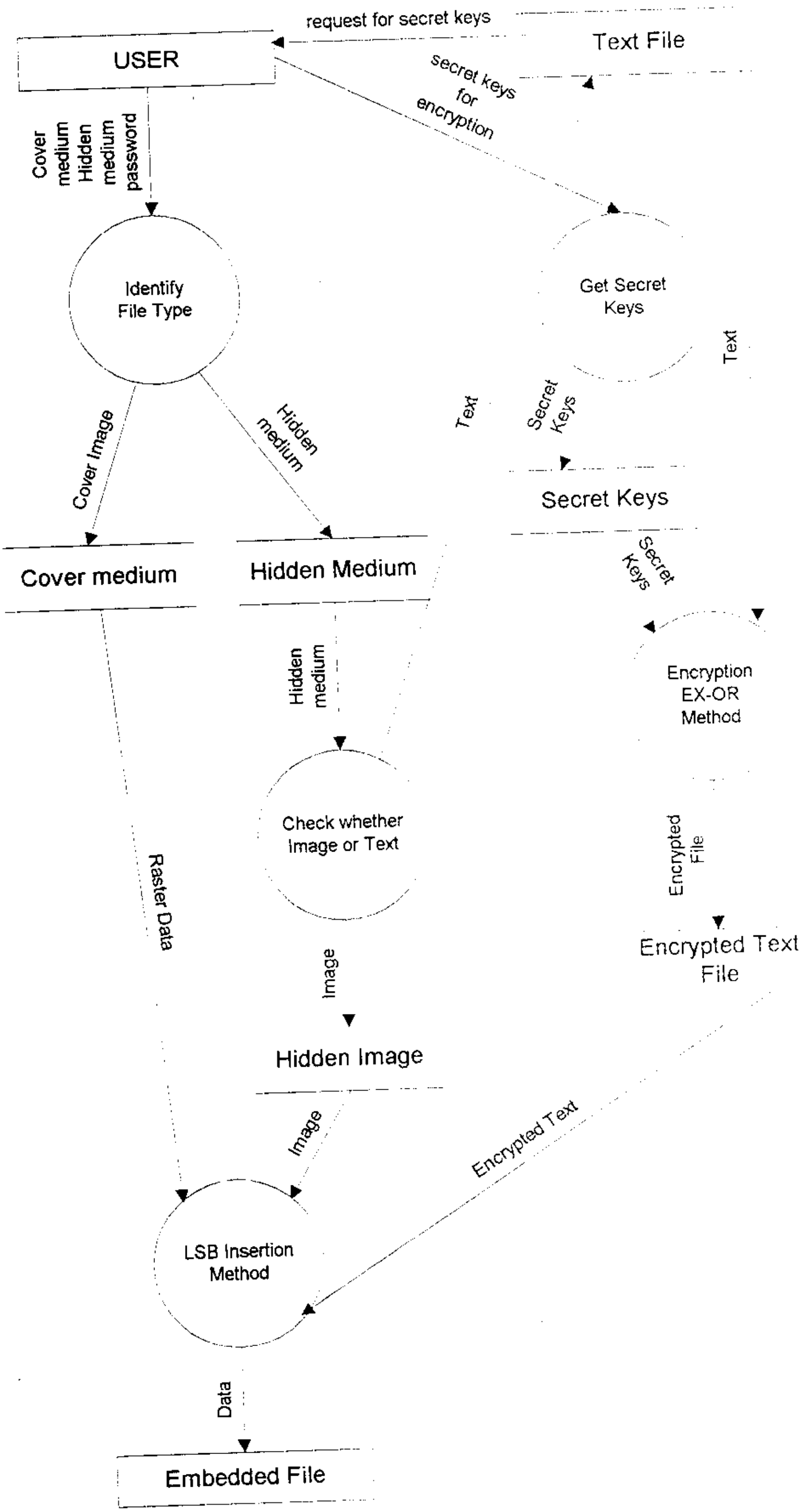
2.6.4 High-Level Data Flow Diagram:

Data Flow Diagram (DFD) is a graphical technique that depicts information flow and the transforms that are applied as data move from input to output. DFD may be used to represent a system or software at any level of abstraction flow and functional details.

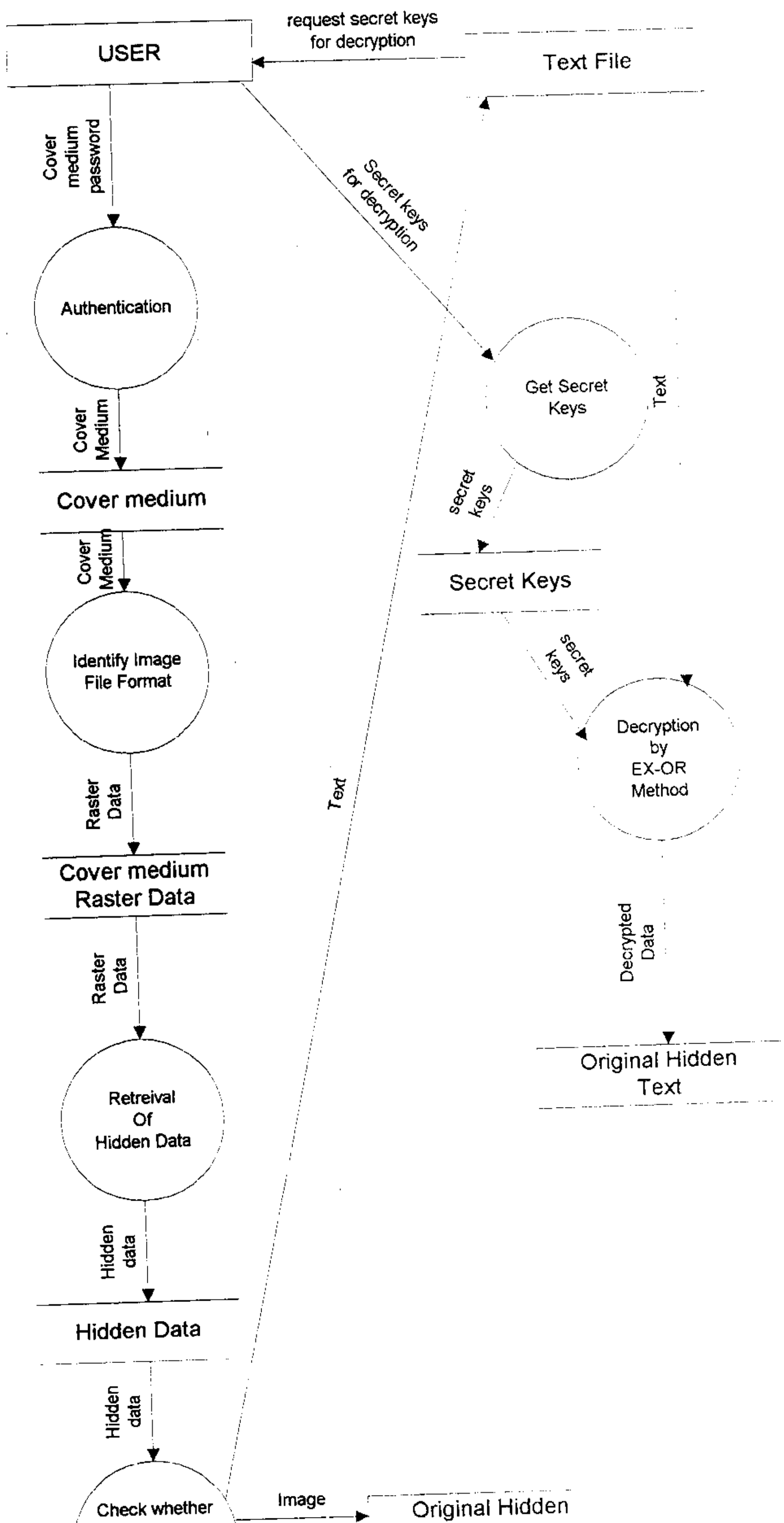
Graphical Notations used in DFD:

Inputs/Outputs to the Software	▭
Processing the inputs	○
Data Flows	→
Data Stores	═

DFD for Hiding the Data in an Image



DFD For Retrieving Data from an Image



2.6.5 Other Requirements:

Operations Required By User:

The user has to enter the path of the image file which is considered to be the cover media, that is the carrier medium, then the user has to enter the path of the text file or the image file that is to be embedded or hidden. If the user prefers to transfer more securely then by selecting the cryption process , the secret keys and the name of a file for storing the crypted message are to be entered. For this entire process of hiding the message a password should be given, which will be required for authentication later during the retrieval process.

The following are the image file format details that were retrieved from the sample images of type BMP, JPG and GIF which were given as input.

Sample Input: Chevy.bmp

Output of Image file details:

```
Version=BM
File Size=2102 bytes
Reserved=0
Data Offset=1078 bytes
Size of header information=40 bytes
Width=32 pixels
Height=32 pixels
Planes=1
Bit Count=8 bits/pixel
Compression=0
Image Size=1024 bytes
X pixels (horizontal pixels)=3790 pixels/meter
Y pixels (vertical pixels)=3800 pixels/meter
Colors Used=0
Colors Important=0
```

Sample Input: Jeep.bmp

Output of Image file details:

```
Version=BM
File Size=1686 bytes
Reserved=0
Data Offset=118 bytes
Size=40 bytes
Width=58 pixels
Height=49 pixels
Planes=1
Bit Count=16
Compression=0
Image Size=1568 bytes
X pixels (horizontal resolution)=3780 pixels/meter
Y pixels (vertical resolution)=3780 pixels/meter
Colors Used=0
Colors Important=0
```

Sample Input: Amex.bmp

Output of Image file details:

```
Version=BM
File Size=120054 bytes
Reserved=0
Data Offset=54 bytes
Size=40 bytes
Width=200 pixels
Height=200 pixels
Planes=1
Bit Count=24bits/pixel
Compression=0
Image Size=120000 bytes
X pixels (horizontal pixels)=2834
Y pixels (vertical pixels)=2834
Colors Used=0
Colors Important=0
```

Sample Input: Ace.gif

Output of Image file details:

```
Version = G I F 8 9 a
GLOBAL DESCRIPTOR
Height =88
Width=104
Global color map Exists
Color Resolution =8 Bits/Pixel
Reserved=0
No .of bits per pixel=8
Background Color=0
Aspect Ratio=1624
Global Map Entries Starting Position: 14
Global Map Entries Ending Position: 787
Local Descriptor Starting Position: 788

LOCAL DESCRIPTOR
Position  X=0
Position  Y=0
Width=88
Height=104
Local Color Map Exists
Non Interlaced Image
Sorted
Reserved
No .of bits per pixel=8
Local Map Entries Starting Position: 799
Local Map Entries Ending Position: 805
Raster Data Starting Position: 806
```

Sample Input: Soccer.jpg

Output of Image file details:

```
SOI: ffd8
APPO Marker Identifier: ffffffe0
Length: 016
File Identifier Mark: 4a4649460
Version: 1.2
Units=1
X Density: 060
Y Density: 060
SOFO Marker Identifier: ffc0
```

```
Length:017
Data Precision: 8
Image Height:070
Image Width:031
No .of Components:3
DRI Marker Identifier: ffdd
Length:04
Restart Interval:04
DHT Marker Identifier :ffc4
Length:163
HT information:1
SOS Marker Identifier: ffda
Length:012
No. of components in scan:3
Image Data Starting Location: 999
```

3. DESIGN DOCUMENT

3.1 EXTERNAL DESIGN SPECIFICATIONS

3.1.1 Logical format of data files

In this software there are ten files they are BMP8.H, BMP16.H, BMP24.H, GIF8.H, JPEG24.H, BCD representation.H , LSBHide.H, LSBRetreive.H, Cryption.H and SAMP.C.

BMP8.H :

This file is used to retrieve the details of a BMP 8 bit image. According to the file format specifications of the BMP image, the details of the image such as its version, file size, reserved space used or not, data offset which gives starting of raster data in bytes, size of header information, width, height, planes, bit count, compressed or not, image size, horizontal pixels/meter, vertical pixels/meter, colors used and colors important are retrieved.

Eg:

Here is the sample output of BMP8.H file which gives the details of a BMP 8-bit image.

```
Version=BM
File Size=2102 bytes
Reserved=0
Data Offset=1078 bytes
Size of header information=40 bytes
Width=32 pixels
Height=32 pixels
Planes=1
Bit Count=8 bits/pixel
Compression=0
Image Size=1024 bytes
X pixels (horizontal pixels)=3790 pixels/meter
Y pixels (vertical pixels)=3800 pixels/meter
Colors Used=0
Colors Important=0
```

BMP16.H :

This file is used to retrieve the details of a BMP 16 bit image. According to the file format specifications of the BMP image, the details of the image such as its version, file size, reserved space used or not, data offset which gives starting of raster data in bytes, size of header information, width, height, planes, bit count, compressed or not, image size, horizontal pixels/meter, vertical pixels/meter, colors used and colors important are retrieved.

Eg:
Here is the sample output of BMP16.H file which gives the details of a BMP 16-bit image.

```
Version=BM
File Size=1686 bytes
Reserved=0
Data Offset=118 bytes
Size=40 bytes
Width=58 pixels
Height=49 pixels
Planes=1
Bit Count=16
Compression=0
Image Size=1568 bytes
X pixels (horizontal resolution)=3780 pixels/meter
Y pixels (vertical resolution)=3780 pixels/meter
Colors Used=0
Colors Important=0
```

BMP24.H :

This file is used to retrieve the details of a BMP 24 bit image. According to the file format specifications of the BMP image, the details of the image such as its version, file size, reserved space used or not, data offset which gives starting of raster data in bytes, size of header information, width, height, planes, bit count, compressed or not, image size, horizontal pixels/meter, vertical pixels/meter, colors used and colors important are retrieved.

Eg:
Here is the sample output of BMP24.H file which gives the details of a BMP 24-bit image.

```
Version=BM
File Size=120054 bytes
Reserved=0
Data Offset=54 bytes
Size=40 bytes
Width=200 pixels
Height=200 pixels
Planes=1
Bit Count=24bits/pixel
Compression=0
Image Size=120000 bytes
X pixels (horizontal pixels)=2834
Y pixels (vertical pixels)=2834
Colors Used=0
Colors Important=0
```

GIF8.H :

This file is used to retrieve the details of a GIF 8 bit image. According to the file format specifications of the GIF image, the details of the image such as its version, global color map entries which includes height, width, number of bits per pixel, color resolution, aspect ratio, global color map entries starting position and global color map entries ending position. And then local descriptor starting position, local color map entries which includes height, width, number of bits per pixel, whether the image is interlaced or non-interlaced image, local color map entries starting position and local color map entries ending position. Finally we get the starting position of raster data.

Eg:
Here is the sample output of GIF8.H file which gives the details of a GIF 8-bit image.

```
Version = G I F 8 9 a
GLOBAL DESCRIPTOR
Height =88
Width=104
Global color map Exists
Color Resolution =8 Bits/Pixel
Reserved=0
No .of bits per pixel=8
Background Color=0
Aspect Ratio=1624
Global Map Entries Starting Position: 14
Global Map Entries Ending Position: 787
```

```
Local Descriptor Starting Position: 788

LOCAL DESCRIPTOR
Position  X=0
Position  Y=0
Width=88
Height=104
Local Color Map Exists
Non Interlaced Image
Sorted
Reserved
No .of bits per pixel=8
Local Map Entries Starting Position: 799
Local Map Entries Ending Position: 805
Raster Data Starting Position: 806
```

JPEG24.H :

This file is used to retrieve the details of a JPEG 24 bit image. According to the file format specifications of the JPEG image, the details of the image such as the details of the start of the image, application marker, start of the frame, define restart interval marker, define Huffman table, start of scan and finally we get the image data.

Eg:
Here is the sample output of JPEG24.H file which gives the details of a JPEG 24 bit image.

```
SOI: ffd8
APPO Marker Identifier: ffffffe0
Length: 016
File Identifier Mark: 4a4649460
Version: 1.2
Units=1
X Density: 060
Y Density: 060
SOFO Marker Identifier: ffc0
Length:017
Data Precision: 8
Image Height:070
Image Width:031
No .of Components:3
DRI Marker Identifier: ffdd
Length:04
Restart Interval:04
DHT Marker Identifier :ffc4
```

Length:163
HT information:1
SOS Marker Identifier: ffda
Length:012
No. of components in scan:3
Image Data Starting Location: 999

BCD representation.H :

This file is required for Lsb insertion method, retrieval process and for cryption process. This file gives the BCD value for each and every character, which is latter used in steganography and cryptography.

LSBHide.H:

This file deals with hiding the data within the cover medium. It checks for the file size of the cover medium to be 8 times greater than the hidden medium. Then only the **hiding process** takes place else an error message will be generated stating that the "Cover medium should be eight times that of Hidden medium". For the hiding process LSB Insertion method is implemented.

LSBRetreive.H:

In this file the **retrieval process** is implemented. This file retrieves the data from the Lsb (Least Significant Bit) of each and every byte. After retrieving, the data will be written into a text file and that the data is retrieved.

Cryption.H :

This file deals with both encryption and decryption by EX-OR method. It encrypts each and every character of the file according to the secret keys given by the user. The user can give any number of secret keys for encryption. Each and every secret key will be EX-OR'ed with each character of the file that is to be encrypted. Decryption is the reverse process of encryption where each and every character of the encrypted file is read and it is EX-OR'ed with the secret keys given by the user. In this EX-OR method during the decryption process the order of the secret keys can vary.

SAMP.C :

This is the main program which includes all the above files. This program deals with the screen design with 'C' and authentication process.

3.2 ARCHITECTURAL DESIGN SPECIFICATIONS:

3.2.1 Parameter Specifications:

The input data consists of the following parameters.

Input Data	Type	Member Variables
Cover medium	char	img[100]
Hidden medium	char	imgt[100]
Overall password	char	overpwd[50]
Secret Keys	char	seck[50]
Output file name of cryption	char	opcryp[100]

Output Data	Type	Member Variables
Hidden medium	FILE	*hidfile
Decrypted file	FILE	*decfile

3.2.2 Functional description:

The "Steganography" software is decomposed into ten modules based on the function they perform. The ten modules are as follows.

- Image file format identification for BMP
- Image file format identification for GIF
- Image file format identification for JPEG
- Cryption
- Data Hiding in BMP image
- Data Retrieval from BMP Image
- Data Hiding in GIF image
- Data Retrieval from GIF Image
- Data Hiding in JPEG image
- Data Retrieval from JPEG Image.

1. Image file format identification for BMP :

Module Used:

Here the Image file format identification for BMP module is used.

Processing Narrative:

This module is used to retrieve the details of a BMP image. According to the file format specifications of the BMP image, the details of the image such as its version, file size, reserved space used or not, data offset which gives starting of raster data in bytes, size of header information, width, height, planes, bit count, compressed or not, image size, horizontal pixels/meter, vertical pixels/meter, colors used and colors important are retrieved.

With this module we come to know the location of raster or pixel data, where the hidden medium is to be hidden.

2. Image file format identification for GIF :

Module Used:

Here the Image file format identification for GIF module is used.

Processing Narrative:

This module is used to retrieve the details of a GIF image. According to the file format specifications of the GIF image, the details of the image such as its version, global color map details, local color map details and finally the starting position of raster data.

The global color map details includes image width, image height, global color map is present or not, color resolution, number of bits per pixel, reserved space used or not, back ground color, aspect ratio, intensity of red, green and blue color.

The local color map details includes horizontal position of image, vertical position of image, width of image, height of image, local color map exists or not, interlaced or non-interlaced image, colors in the image are sorted or not, reserved space used or not, number of bits per pixel, intensity of red, green and blue color.

Then finally we get the position from where the raster or pixel data starts.

3. Image file format identification for JPEG :

Module Used:

Here the Image file format identification for JPEG module is used.

Processing Narrative:

This module is used to retrieve the details of a JPEG image. According to the file format specifications of the JPEG image, the details of the image such as its version, units for horizontal and vertical density whether inches or centimeters, then value of horizontal and vertical densities and file length.

Then finally we get the position from where the raster or pixel data starts.

4. Cryption :

Module Used:

Here the Cryption module is used.

Processing Narrative:

This module is used to encrypt and decrypt the hidden file. Here cryption by EX-OR method is used. In this crypting technique, encryption is done according to the users secret keys, which are got as input from the user and then the message is encrypted. The encrypted message will be hidden according to the concept of Steganography .

In the case to decrypt an encrypted file the user should provide the secret keys that where entered during the encryption process. Once secret keys are entered the decryption is done and that the original message will be obtained.

5. Data Hiding in BMP Image :

Module Used:

Here the Data Hiding in BMP image module is used.

Processing Narrative:

This module deals with hiding the data using Lsb Insertion method. As the user enters the cover medium and hidden medium as input, here it checks whether the cover medium is eight times as that of the hidden medium, if so then the hiding process is done as per the Lsb Insertion method else an error message is generated as "Cover medium should be eight times that of Hidden medium". For hiding the data within the cover medium the location of the raster data is required which will be obtained from Image file format identification for BMP module and in that raster data the hiding process takes place.

6.Data Retrieval from BMP Image:

Module Used:

Here the Data Retrieval from BMP image module is used.

Processing Narrative:

This module deals with retrieving the data using retrieval process as described earlier. As the user enters the cover medium as input, and clicks the retrieve button, the authentication process takes place. Then the data that resides in the Lsb of raster data of the image is retrieved as per the retrieval process. For retrieving the data within the cover medium the location of the raster data is required which will be obtained from Image file format identification for BMP module and in that raster data the retrieval process takes place.

7. Data Hiding in GIF Image :

Module Used:

Here the Data Hiding in GIF image module is used.

Processing Narrative:

This module deals with hiding the data using Lsb Insertion method. As the user enters the cover medium and hidden medium as input, here it checks whether the cover medium is eight times as that of the hidden medium, if so then the hiding process is done as per the Lsb Insertion method else an error message is generated as "Cover medium should be eight times that of Hidden medium". For hiding the data within the cover medium the location of the raster data is required which will be obtained from Image file format identification for GIF module and in that raster data the hiding process takes place.

8. Data Retrieval from GIF Image:

Module Used:

Here the Data Retrieval from GIF image module is used.

Processing Narrative:

This module deals with retrieving the data using retrieval process as described earlier. As the user enters the cover medium and password as input, and clicks the retrieve button, the authentication process takes place. Then the data that resides in the Lsb of raster data of the image is retrieved as per the retrieval process. For retrieving the data within the cover medium the location of the raster data is required which will be obtained from Image file format identification for GIF module and in that raster data the retrieval process takes place.

9. Data Hiding in JPEG Image :

Module Used:

Here the Data Hiding in JPEG image module is used.

Processing Narrative:

This module deals with hiding the data using Lsb Insertion method. As the user enters the cover medium and hidden medium as input, here it checks whether the cover medium is eight times as that of the hidden medium, if so then the hiding process is done as per the Lsb Insertion method else an error message is generated as "Cover medium should be eight times that of Hidden medium". For hiding the data within the cover medium the location of the raster data is required which will be obtained from Image file format identification for JPEG module and in that raster data the hiding process takes place.

10. Data Retrieval from JPEG Image:

Module Used:

Here the Data Retrieval from JPEG image module is used.

Processing Narrative:

This module deals with retrieving the data using retrieval process as described earlier. As the user enters the cover medium and password as input, and clicks the retrieve button, first the authentication process takes place. Then the data that resides in the Lsb of raster data of the image is retrieved as per the retrieval process. For retrieving the data within the cover medium the location of the raster data is required which will be obtained from Image file format identification for JPEG module and in that raster data the retrieval process takes place.

3.3 Packing Specification:

The organization has to run the executable program to make use of this tool. There is no special packing done for this software.

4. TEST PLANS

1. **Type of test** : Functional test

Machine Configuration:

Processor: Pentium III 550 MHz

Floppy Disk Drive: 1.44 MB

Hard Disk: 20GB

System RAM: 64MB

Test assumption: Invalid input

Exact Test Stimuli:

If the user enters an input that is the cover medium, image file and the hidden medium image file or the text file that doesn't exists in the current location where the software is loaded then an error message should occur such that the specified file does not exist and to enter the correct file name.

Expected outcome:

The message is displayed as "File Doesn't Exist. Enter Correct File name or Verify whether File exists in current location".

2. **Type of test** : Performance test

Machine Configuration:

Processor: Pentium III 550 MHz

Floppy Disk Drive: 1.44 MB

Hard Disk: 20GB

System RAM: 64MB

Test assumption: If the cover medium (image file) size is not eight times as that of hidden medium (image or text file) size.

Exact Test Stimuli:

As the user enters the cover medium and the hidden medium file names. The software program checks whether the cover medium (image file) is eight times as that of hidden medium (image or text file) so that the hidden medium can be embedded within the cover medium.

Expected outcome:

The message is displayed as "Cover medium should be eight times that of Hidden medium".

3. **Type of test** : Compliance test

Machine Configuration :

Processor: Pentium III 550 MHz
Floppy Disk Drive: 1.44 MB
Hard Disk: 20GB
System RAM: 64MB

Test assumption: If wrong secret keys are entered for decryption.

Exact Test Stimuli:

When the user enters wrong secret keys for decryption, a message will be displayed asking to enter correct secret keys. The secret keys that were entered by the user during the encryption, the same secret keys should be entered by the user for decryption not necessarily in the same order.

Expected outcome:

The message is displayed as "Enter the correct secret keys which were entered during encryption".

4. **Type of test** : Compliance test

Machine Configuration :

Processor: Pentium III 550 MHz

Floppy Disk Drive: 1.44 MB

Hard Disk: 20GB

System RAM: 64MB

Test assumption: If wrong password is entered for carrying out the Steganography process.

Exact Test Stimuli:

When the user enters wrong password for retrieving the hidden medium, a message will be displayed asking to enter correct password as such which was entered during the hiding process.

Expected outcome:

The message is displayed as "Enter the correct password for retrieving the information".

5. PROJECT LEGACY

5.1 PROJECT DESCRIPTION

The project “Steganography” deals with embedding an image file within an image file or embedding a text file within an image file. In this case the software will be designed such that the user selects the two files, an image and a text file, where the image is the cover media, in which the text file is to be embedded directly, or it will be embedded after encrypting the text file. On the other hand two image files can be embedded and transferred. Steganography concept by itself is secure as it does not even show the existence of the information within the other medium. As in the case to provide more security to the private information we combine the concept of Cryptography along with the Steganography. This concept is very much useful for transfer of secure information.

5.2 INTIAL EXPECTATIONS

The objective of "Steganography" software is to hide the messages in a way that it does not allow any enemy to even detect that there is a second secret message present. The Steganography is to be done for Images and Text. Such that hiding an Image within another image and hiding the text within another image. In the second case that is while hiding the text within another image, in order to provide more privacy the text is encrypted and then hidden within the image. Then while retrieving the hidden text from the image, the text that was encrypted has to be decrypted and that the original text will be obtained. Here the images that we are dealing with are BMP (8,16 and 24 bit image), GIF (8 bit image) and JPEG (24 bit image).

5.3 SOLUTION STRATEGY:

The problem was approached in a step by step fashion. First and foremost the technical concept of Steganography was learnt which hiding and retrieving processes. Then the image file formats specifications of BMP (8,16 and 24 bit image), GIF (8 bit image) and JPEG (24 bit image) were learnt. Then analysis was done on problem definition given by the Project Manager. Then the rough draft is made on the analysis and finally ends up in a solution by breaking down the defined problem into ten modules.

First the cover medium and hidden medium is got as input from the user and is checked whether the size of the cover medium is eight times that of the hidden medium. If both the cover medium and hidden medium are image files, then image file format identification is done for both cover and hidden medium to check whether it is a BMP (8,16 and 24 bit image), GIF (8 bit image) or JPEG (24 bit image), if the image file format is other than the specified then an error message is generated such that "Supports only BMP (8,16 and 24 bit image), GIF (8 bit image) or JPEG (24 bit image)". Then according to image file format given as input the hidden medium is hidden within the cover medium which is dealt by one of the modules such as Data Hiding in BMP image, Data Hiding in GIF image or Data Hiding in JPEG image module as per the user's input. On the other hand if the hidden medium is a text file and the cover medium is an image file then image file format identification is done as before. As per the user's wish the text file can be encrypted in order to maintain more secrecy and then the encrypted file is to be hidden. In such a case the user has to enter the secret keys for encryption and the file name in which the encrypted message will be stored which will be dealt in cryption module. Then this encrypted file will be hidden within the image file which will be dealt by one of the modules such as Data Hiding in BMP image, Data Hiding in GIF image or Data Hiding in JPEG image module as per the user's input.

Then in order retrieve the hidden medium from the cover medium (image file), the image file format identification is done for the cover medium to check whether it is a BMP (8,16 and 24 bit image), GIF (8 bit image) or JPEG (24 bit image), if the image file format is other than the specified then an error message is generated such that "Supports only BMP (8,16 and 24 bit image), GIF (8 bit image) or JPEG (24 bit image)". Then hidden medium is retrieved from the cover medium which will be dealt by Data Retrieval from BMP image, Data Retrieval from GIF image or Data Retrieval from JPEG image module as per the user's input which is the cover medium. Then if the hidden medium is a text file then the decryption is to be done for which the user has to enter the secret keys and the file name in which decrypted information is to be stored which will be dealt in cryption module.

5.4 SOFTWARE PRODUCT FEATURES:

The function of the product is to embed two files such that embedding an image file within an image file or embedding a text file within an image file. In this case the software will be designed such that the user selects the two files, an image and a text file, where the image is the cover media, in which the text file is to be embedded directly, or it will be embedded after encrypting the text file. On the other hand two image files can embedded and transferred. Steganography concept by itself is secure as it does not even show the existence of the information within the other medium. As in the case to provide more security to the private information we combine the concept of Cryptography along with the Steganography. In this crypting technique, encryption is done according to the users secret keys, which are to be got as input from the user and then the message is encrypted. The encrypted message will be hidden according to the concept of Steganography and then the entire Stego media, that is Carrier media with hidden message will be transmitted.

To retrieve the embedded file, the user selects the stego media and then when the user clicks the retrieve button the software asks for the password for authentication. This password should match with the password which was given by the user during the hiding process. After authentication is successful the embedded file is retrieved, if in the case if the embedded file is an encrypted file the user should provide the secret keys that where entered during the encryption process. Once secret keys are entered the decryption is done and that the original message is retrieved.

5.5 ACCEPTANCE CRITERIA:

The software has to embed the image or the text file within the image file. If the hidden medium is a text file according to user's wish encryption will be done for which the secret keys and the file name in which the encrypted message is to be stored will be got as input form the user. Then this encrypted file will be hidden within the cover medium. If the hidden medium is an image file it will be hidden directly within the cover medium. Then the software should retrieve the data from the cover medium during the retrieval process and if the retrieved file is an encrypted text file then decryption is to be done for which the user has to enter the secret keys as input.

5.6 PROJECT PLAN:

5.6.1 Life Cycle Mode:

The Spiral Model is proposed to be the life cycle mode followed while developing the product software. It provides the potential for the rapid development of incremental version of the software. The software is developed in the series of incremental releases. The Spiral Model has six tasks region.

Task Region 1:

- Terminology: Communication with leader
- Milestones: Nov 20th - Nov 22nd
- Work Product: The project leader defines about Steganography and Cryption. Gathered notes from the websites and some books related to Steganography and its use. The "Steganography" software should be able to identify the image file formats such as BMP, GIF and JPEG and hide the images or the text given by the user. If it is a text file as according to the wish of the user it should encrypt the text file before hiding and then hide the encrypted file within the cover medium (image

file) given by the user. Then it should also be able to retrieve the message from the cover medium (image file) given by the user as input and then decrypt the file if required.

Task Region 2:

- Terminology: Planning
- Milestones: Nov 23rd – Nov 28th
- Work Product: Analysis of the software product definition. What function the software product has to perform, Processing environment, Software features, Programming language to be used are all decided in this stage.

Task Region 3:

- Terminology : Risk Analysis
- Milestones: Nov 29th – Dec 3rd
- Work Product:

Technical Risk:

In order to implement steganography in BMP, GIF and JPEG the details of the image file formats should be known. Then the cryption method is to be decided which is to be combined with steganography. The images which are of less than 8 bits per pixel such as 4 bits per pixel and monochrome images do not support steganography, as changes to the bits in these images will result in drastic change in color and image quality.

Managerial Risk:

For every module time limits are set to be fixed for its completion. The project duration is between Nov 20th - Feb 29th. Within the given time slot for each module, it has to be completed. The modules cannot be postponed because output of some modules becomes the input for other module. If there is a delay in any one of the modules there is a delay in completion of the whole project. Therefore the modules have to be completed in the appropriate time as per schedule.

Task Region 4:

- Terminology: Engineering
- Milestones: SRS Document Dec 3rd – Dec 6th
- Work Product: Based on the needs, the software requirement specification is prepared.

- Milestones: Design Document Dec 7th – Dec 18th
- Work Product: Based on the needs of the customer, the Design document is prepared. Designing plays an important role in coding. Once the design is framed well, the programmer can start the coding very easily. Design document includes external design specification, Architectural design overview and Detailed design Specification.

Task Region 5:

Terminology: Construction and Release.

Milestones	**Work Product**
Dec 19th – Dec 26th	Coding for Image file format identification for BMP module.
Dec 27th – Jan 2nd	Coding for Image file format identification for GIF module.

Jan 3rd – Jan 7th	Coding for Image file format identification for JPEG module.
Jan 8th – Jan 11th	Coding for Cryption module
Jan 12th – Jan 22nd	Coding for Data Hiding in BMP image module.
Jan 23rd – Jan 27th	Coding for Data Retrieval from BMP image module.
Jan 28th – Jan 31st	Coding for Data Hiding in GIF image module.
Feb 1st – Feb 3rd	Coding for Data Retrieval from GIF image module.
Feb 4th – Feb 10th	Coding for Data Hiding in JPEG image module.
Feb 11th – Feb 15th	Coding for Data Retrieval from JPEG image module.
Feb 16th – Feb 18th	Preparing User Manual
Feb 18th – Feb 22nd	Preparing Test Plan
Feb 23rd –Feb 24th	Installing EXE file
Feb 25th-Feb 27th	Product Evaluation

Task Region 6:

- Terminology: Leader Evaluation
- Milestones: Feb 25th – Feb 27th
- Work Product: On seeing the execution, the project leader gave the feedback. His feedback was the product works perfectly and it can be extended in future for most of the image file formats.

5.6.2 Team Structure :

The "Steganography" software is used for hiding and retrieving the information from the image file as per the user's request.

The project consists of one member under the guidance of the Project leader.

5.6.3 Development Schedule :

In order to complete the project in a given time, the development schedule is framed and based on the time slots, the software product is developed. The development schedule consists of Milestones and Reviews.

Milestones	**Reviews**
Nov 22nd - Product Definition	A rough draft is made to Product Definition and definitions are reviewed.
Nov 26th - Product Analysis	A rough draft is made to Product Analysis and the review is made on

	the analysis to do step by step fashion.
Nov 28th - Programming language	The Programming Language is decided.
Nov 2nd - Risk Analysis	A rough draft is made to Risk Analysis. The draft was reviewed and there was two introduction types of risks like Technical Risk and Managerial Risk.
Dec 6th - SRS	SRS general formats are reviewed.
Dec 18th - Design Documentation	Design Document was reviewed.
Dec 26th - Coding for Image file format identification for BMP module	The reviews were made on the logic.
Jan 2nd - Coding for Image file format identification for GIF module.	The reviews were made on the logic.
Jan 7th - Coding for Image file format identification for JPEG module.	The reviews were made on the logic.
Jan 11th - Coding for Cryption module.	The reviews were made on the logic.
Jan 22nd - Coding for Data Hiding in BMP image module.	The reviews were made on the logic.
Jan 27th - Coding for Data Retrieval from BMP image module.	The reviews were made on the logic

Jan 31st - Coding for Data Hiding in GIF image module.	The reviews were made on the logic.
Feb 3rd - Coding for Data Retrieval from GIF image module.	The reviews were made on the logic.
Feb 10th - Coding for Data Hiding in JPEG image module	The reviews were made on the logic.
Feb 15th - Coding for Data Retrieval from JPEG image module.	The reviews were made on the logic.
Feb 18th - Prepare user manual	Reviews were made on the rough draft of the user manual.
Feb 22nd - Test Plans	After doing the various test plans, reviews were made on rough draft of the test plans.
Feb 24th - Installing Exe program	No Reviews
Feb 27th - Full project demonstration	No Reviews

5.6.4 Programming Language :

'C' is the programming language that is been used in the software "Steganography".

'C' is a general purpose structured programming language that is powerful, efficient and compact . 'C' combines the features of a high-level language with the elements of the assembler and is thus, close to both man and machine.

Turbo C Compiler is the most widely used professional software development tool on the micro computers.

Features of the 'C' Language:

C language as such offers only a handful of functions which form the core of the language, rest of the functions available in libraries are developed using the core functions offered by the language as building blocks. This feature expands the scope and the power of the language. This leads to functionally cohesive modules and, therefore, re-usable code.

Thus highly independent functions can be written and stored in a library containing other functions to be used as when necessary.

Thus C language makes it possible to use the language for systems programming, like development of compilers, interpreters, operating systems, graphics and general utilities, and also for a host of applications in the commercial environment.

5.6.5 Documents to be prepared:

It was suggested the following documents are to be prepared during the time of the project.

- A System Definition consisting of a Product Definition and a Project Plan.
- A Software Requirement Specification.
- A Design document consisting of external design and architectural design
- A Test Plan
- A User's manual
- A Project Legacy document.

5.6.6 Manner of Demonstration:

Reviews:

Every week on Friday the finished modules are explained to the project leader, reviewed and inputs and outputs are verified.

Documents:

Draft of every document is reviewed by the project leader before it is finalized. If there are any changes to the draft they are incorporated in the document.

Product:

Demo of the each module is given to the project leader as and when the module is completed. If any changes are required, they are incorporated in the module after the review.

5.7 CURRENT STATUS OF THE PROJECT

The "Steganography" is able to do all the operations like image file format identification, data hiding within the image, cryption and data retrieval from the image. Here the cover medium is an image file and hidden medium can be an image file or text file. The image files that are supported by this software are BMP (8,16 and 24 bit image), GIF (8 bit image) and JPEG (24 bit image).

5.8 ACTIVITIES / TIME LOG(S)

Time Logs	**Activities**
Nov 22nd	Product Definition

Nov 26th	Product Analysis
Nov 28th	Programming language
Nov 2nd	Risk Analysis
Dec 6th	SRS
Dec 18th	Design Documentation
Dec 26th	Coding for Image file format identification for BMP module.
Jan 2nd	Coding for Image file format identification for GIF module.
Jan 7th	Coding for Image file format identification for JPEG module.
Jan 11th	Coding for Cryption module.
Jan 22nd	Coding for Data Hiding in BMP image module.
Jan 27th	Coding for Data Retrieval from BMP image module.
Jan 31st	Coding for Data Hiding in GIF image module.
Feb 3rd	Coding for Data Retrieval from GIF image module.

Feb 10th	Coding for Data Hiding in JPEG image module.
Feb 15th	Coding for Data Retrieval from JPEG image module.
Feb 18th	Prepare User Manual
Feb 22nd	Test Plans
Feb 24th	Installing Exe program
Feb 27th	Full project demonstration

5.9 TECHNICAL LESSONS LEARNED

Many technical lessons were learned while working in this project. Out of many techniques the best one is selected, learnt and implemented in the case of Problem analysis and it's solving.

Based on the problem approach and its solution the software is selected. The software used is 'C'. 'C', plays a main role in hiding and retrieval of information.

Many methods regarding the manipulation of bits were learnt to approach the problem. Some websites and books were also referred to know about the Image file formats, Cryption and Steganography.

5.10 MANAGERIAL LESSONS LEARNT

In this organization apart from the technical lessons, managerial lessons were also learnt, that includes - How to approach the problem patiently and how to discuss about the evaluation of the problem and express my solution regarding the problem in the conference. The way of communicating with my Project Leader and programmers and asking them suggestions and advises in some tough times while programming. The time slots were framed by me and the Project leader to finish each and every part of the module. Suppose if there is a delay in coding for a particular module, the time for the very next modules timing should be adjusted in order to finish the project in a given time slot. As and when the modules are finished the time taken to finish is noted in a sheet of paper and it is compared with the original time slot.

5.11 RECOMMENDATIONS TO FUTURE PROJECTS

In this organization the projects are mainly concerned with developing software for embedded systems and images. They develop software tools for Image compression and decompression. Many projects can be done in future on images.

6. CONCLUSION

The "Steganography" is the software that helps in hiding the messages safely and transmitting to the destination. It is more powerful than Cryptography such that when encryption is done to a message it shows the existence of some message. In the case of Steganography it even hides the existence of the information within another medium. It can also be applied to reduce the network traffic and thus the consumption of network bandwidth. On an average it only alters 50% of the bits. It works best with 24 bit cover images and it is undetectable to human eye.

Steganography and Cryptography can be combined to maintain the secrecy of the message effectively.

The software with some modifications can be applied to all image formats from 8 bits/pixel onwards. Steganography will not suit for monochrome and less than 8 bits/pixel images, as altering some of the bits in these images will cause changes in the color and image quality, which will lead to suspicion that some message is hidden.

The programming style used adds to the flexibility of the system and allows easy modification to the existing system. Appropriate comments and suggestions are specified throughout the code, which make the code easily understandable. Clear documentation and supporting documents along with appropriate diagrams have been produced which aids in understanding the overall system architecture.

Apart from these I gained a lot of technical knowledge, industrial mannerism and behavior. I was able to understand and learn more about the real world project life - cycle.

REFERENCES

Books:

- Neil F. Johnson, Zoran Duric, Sushil Jajodia, **“Information Hiding: Steganography and Watermarking - Attacks and Countermeasures”**, Kluwer Academic Publishers, 2000.
- Charles H.Roth, Jr. University Of Texas At Austin, **"Fundamentals Of Logic Design"**, Jaico Publishing House 2002.
- Yashvant Kanetkar, **“Graphics using ‘C’ ”**
- Roger S. Pressman, **"Software Engineering"**, McGraw-Hill International Edition 2001.

Web Sites:

www.google.com

http://students.washington.edu/tjarvis2/steganography/tracys.steganography.introduction.html (November 2003).

http://www.cs.uct.ac.za/courses/CS400W/NIS/papers99/dsellars/stego.html (January 2004).

http://www.jjtc.com/Steganography.html (January 2004).

www.computerworld.com

A. USER'S MANUAL

A.1 INTRODUCTION:

A.1.1 Product rational and overview:

This Product is mainly concerned for transferring the private information and to reduce the network bandwidth.

The objective of "Steganography" software is to hide the messages in a way that it does not allow any enemy to even detect that there is a second secret message present. It deals with Image file format identification of BMP,GIF and JPEG, Cryption, Data Hiding in BMP image, Data Retrieval from BMP Image, Data Hiding in GIF image, Data Retrieval from GIF Image, Data Hiding in JPEG image, Data Retrieval from JPEG Image. The image file formats are to be identified according to the file format specifications of BMP, GIF and JPEG. After the image file format identification we get the details of the image such as its file size, height, width, number of bits per pixel, resolution, the colors that are used and from which byte the raster or the pixel data starts, where we are going to hide the message.

First the cover medium and hidden medium is got as input from the user and is checked whether the size of the cover medium is eight times that of the hidden medium. If both the cover medium and hidden medium are image files, then image file format identification is done for both cover and hidden medium to check whether it is a BMP (8,16 and 24 bit image), GIF (8 bit image) or JPEG (24 bit image), if the image file format is other than the specified then an error message is generated such that "Supports only BMP (8,16 and 24 bit image), GIF (8 bit image) or JPEG (24 bit image)". Then according to image file format given as input the hidden medium is hidden within the cover medium which is dealt by one of the modules such as Data Hiding in BMP image, Data Hiding in GIF image or Data Hiding in JPEG image module as per the user's input. On the other hand if the hidden medium is a text file and the cover medium is an image file then image file format identification is done as before. As per the user's wish the text file can be encrypted in order to maintain more secrecy and then the encrypted file is to be hidden. In such a case the user has to enter the secret keys for encryption and the file name in which the encrypted message will be stored which will be dealt in cryption module. Then this encrypted file will be hidden within the image file which will be dealt by one of the modules such as Data Hiding in BMP image, Data Hiding in GIF image or Data Hiding in JPEG image module as per the user's input.

A.1.2 Terminology:

The software Steganography, is such that embedding an image file within an image file or embedding a text file within an image file. In this case the software will be designed such that the user selects the two files, an image and a text file, where the image is the cover media, in which the text file is to be embedded directly, or it will be embedded after encrypting the text file. On the other hand two image files can embedded and transferred. Steganography concept by itself is secure as it does not even show the existence of the information within the other medium. As in the case to provide more security to the private information we combine the concept of Cryptography along with the Steganography.

Definitions:

- Steganography- Steganography is a means of storing the information in a way that hides private information and even the existence of the information within the other medium.
- Cover media- Cover media is the carrier medium, like text, image, audio, video.
- Secret message- Secret message is the private message that is to be hidden inside the Cover media.
- Stego media- Carrier with hidden message.

- Pixel- An abbreviation of the term 'picture element.' A pixel is the smallest picture element of a digital image.
- Resolution- The number of pixels in a graphic or screen in the horizontal and vertical dimensions, which governs the level of fine detail images can have.
- Cryptography-A system for encrypting and decrypting data is a cryptosystem.
- Encryption- Encryption is a process of translating a message, called the Plaintext, into an encoded message, called the Ciphertext.
- Decryption- Decryption is the reverse process to Encryption. Frequently, the same Cipher is used for both Encryption and Decryption. While Encryption creates a Ciphertext from a Plaintext, Decryption creates a Plaintext from a Ciphertext.
- Ciphertext- Ciphertext is encoded text, after it has been passed through an Encryption algorithm. It is the product of Plaintext after Encryption.
- Plaintext- Plaintext is an unencrypted message, before it is passed through an Encryption algorithm (Cipher). It is used to create a Ciphertext.
- Cipher- A Cipher is a computer software algorithm used for Encryption.
- Secret Key- The key that is used for both encryption and decryption
- Network Bandwidth- Refers to the data rate supported by a network connection or interface. One most commonly expresses bandwidth in terms of bytes per second (Bps).

A.1.3 Basic Features:

The function of the product is to embed two files such that embedding an image file within an image file or embedding a text file within an image file. In this case the software will be designed such that the user selects the two files, an image and a text file, where the image is the cover media, in which the text file is to be embedded directly, or it will be embedded after encrypting the text file. On the other hand two image files can embedded and transferred. Steganography concept by itself is secure as it does not even show the existence of the information within the other medium. As in the case to provide more security to the private information we combine the concept of Cryptography along with the Steganography. In this crypting technique, encryption is done according to the users secret keys, which are to be got as input from the user and then the message is encrypted. Γhe encrypted message will be hidden according to the concept of Steganography and then the entire Stego media, that is Carrier media with hidden message will be transmitted.

To retrieve the embedded file, the user selects the stego media and then when the user clicks the retrieve button the software asks for the password for authentication. This password should match with the password, which was given by the user during the hiding process. After authentication is successful the embedded file is retrieved, if in the case if the embedded file is an encrypted file the user should provide the secret keys that where entered during the encryption process. Once secret keys are entered the decryption is done and that the original message is retrieved.

A.2 GETTING STARTED:

A.2.1.Loading of files:

In this software there are eight files they are BMP8.H, BMP16.H, BMP24.H, GIF8.H, JPEG24.H, LSBHide.H, LSBRetreive.H, Cryption.H and SAMP.C

Here follows the brief description of each and every file and how they are to be loaded and used.

BMP8.H File:

BMP8.H file is used to identify the image file format and that it finds out the raster or pixel data within which it hides the hidden medium.

BMP16.H File:

BMP16.H file is used to identify the image file format and that it finds out the raster or pixel data within which it hides the hidden medium.

BMP24.H File :

BMP24.H file is used to identify the image file format and that it finds out the raster or pixel data within which it hides the hidden medium.

GIF8.H File :

GIF8.H file is used to identify the image file format and that it finds out the raster or pixel data within which it hides the hidden medium.

JPEG24.H File :

JPEG24.H file is used to identify the image file format and that it finds out the raster or pixel data within which it hides the hidden medium.

BCD representation.H File :

This file is required for Lsb insertion method, retrieval process and for cryption process. This file gives the BCD value for each and every character, which is latter used in steganography and cryptography.

LSBHide.H File :

This file does the hiding process within the raster data as per the LSB insertion method and that it hides the hidden medium within the cover medium.

LSBRetreive.H File :

This file does the retrieval process from the raster data and that it retrieves the hidden medium from the cover medium.

Cryption.H File:

This file is used to encrypt and decrypt the file that is given as input by the user.

SAMP.C File :

This file is the main file which includes all the above files, this deals with screen design and validation.

Executing operation:

First the cover medium and hidden medium is got as input from the user and is checked whether the size of the cover medium is eight times that of the hidden medium. If both the cover medium and hidden medium are image files, then image file format identification is done for both cover and hidden medium to check whether it is a BMP (8,16 and 24 bit image), GIF (8 bit image) or JPEG (24 bit image), if the image file format is other than the specified then an error message is generated such that "Supports only BMP (8,16 and 24 bit image), GIF (8 bit image) or JPEG (24 bit image)". Then according to image file format given as input the hidden medium is hidden within the cover medium which is dealt by one of the modules such as Data Hiding in BMP image, Data Hiding in GIF image or Data Hiding in JPEG image module as per the user's input. On the other hand if the hidden medium is a text file and the cover medium is an image file then image file format identification is done as before. As per the user's wish the text file can be encrypted in order to maintain more secrecy and then the encrypted file is to be hidden. In such a case the user has to enter the secret keys for encryption and the file name in which the encrypted message will be stored which will be dealt in cryption module. Then this encrypted file will be hidden within the image file which will be dealt by one of the modules such as Data Hiding in BMP image, Data Hiding in GIF image or Data Hiding in JPEG image module as per the user's input.

Then in order retrieve the hidden medium from the cover medium (image file), the image file format identification is done for the cover medium to check whether it is a BMP (8,16 and 24 bit image), GIF (8 bit image) or JPEG (24 bit image), if the image file format is other than the specified then an error message is generated such that "Supports only BMP (8,16 and 24 bit image), GIF (8 bit image) or JPEG (24 bit image)". Then hidden medium is retrieved from the cover medium which will be dealt by Data Retrieval from BMP image, Data Retrieval from GIF image or Data Retrieval from JPEG image module as per the user's input which is the cover medium. Then if the hidden medium is a text file then the decryption is to be done for which the user has to enter the secret keys and the file name in which decrypted information is to be stored which will be dealt in cryption module.

A.2.2 Sample Run:

The following are the image file format details that were retrieved from the sample images of type BMP, JPG and GIF which were given as input.

Sample Input: Toyota.bmp

Output of Image file details:

```
Version=BM
File Size=2102 bytes
Reserved=0
Data Offset=1078 bytes
Size of header information=40 bytes
```

```
Width=32 pixels
Height=32 pixels
Planes=1
Bit Count=8 bits/pixel
Compression=0
Image Size=1024 bytes
X pixels (horizontal pixels)=3790 pixels/meter
Y pixels (vertical pixels)=3800 pixels/meter
Colors Used=0
Colors Important=0
```

Sample Input: Driftwood.bmp

Output of Image file details:

```
Version=BM
File Size=1686 bytes
Reserved=0
Data Offset=118 bytes
Size=40 bytes
Width=58 pixels
Height=49 pixels
Planes=1
Bit Count=16
Compression=0
Image Size=1568 bytes
X pixels (horizontal resolution)=3780 pixels/meter
Y pixels (vertical resolution)=3780 pixels/meter
Colors Used=0
Colors Important=0
```

Sample Input: Advus.bmp

Output of Image file details:

```
Version=BM
File Size=120054 bytes
Reserved=0
Data Offset=54 bytes
Size=40 bytes
Width=200 pixels
Height=200 pixels
Planes=1
Bit Count=24bits/pixel
Compression=0
```

```
Image Size=120000 bytes
X pixels (horizontal pixels)=2834
Y pixels (vertical pixels)=2834
Colors Used=0
Colors Important=0
```

Sample Input: Sanfran.gif

Output of Image file details:

```
Version = G I F 8 9 a
GLOBAL DESCRIPTOR
Height =88
Width=104
Global color map Exists
Color Resolution =8 Bits/Pixel
Reserved=0
No. of bits per pixel=8 Bits
Background Color=0
Aspect Ratio=1624
Global Map Entries Starting Position: 14
Global Map Entries Ending Position: 787
Local Descriptor Starting Position: 788

LOCAL DESCRIPTOR
Position X=0
Position Y=0
Width=88
Height=104
Local Color Map Exists
Non Interlaced Image
Sorted
Reserved
No .of bits per pixel=8
Local Color Map Exists
Local Map Entries Starting Position:799
Local Map Entries Ending Position:805
Raster Data Starting Position:806
```

Sample Input: Ferrari.jpg

Output of Image file details:

```
SOI: ffd8
APPO Marker Identifier: ffffffe0
Length: 016
File Identifier Mark: 4a4649460
Version: 1.2
Units=1
X Density: 060
Y Density: 060
SOFO Marker Identifier: ffc0
Length:017
Data Precision: 8
Image Height:070
Image Width:031
No .of Components:3
DRI Marker Identifier: ffdd
Length:04
Restart Interval:04
DHT Marker Identifier :ffc4
Length:163
HT information:1
SOS Marker Identifier: ffda
Length:012
No. of components in scan:3
Image Data Starting Location: 999
```

A.3 MODES OF OPERATION:

The " Steganography", comprises of ten modules. They are

- Image file format identification for BMP
- Image file format identification for GIF
- Image file format identification for JPEG
- Cryption
- Data Hiding in BMP image
- Data Retrieval from BMP Image
- Data Hiding in GIF image
- Data Retrieval from GIF Image

- Data Hiding in JPEG image
- Data Retrieval from JPEG Image.

There are two modes of operation they are

- Keyboard
- Mouse

Keyboard:

With the help of the keyboard the user can enter the inputs such as the file names of cover medium, hidden medium, the output file name, secret keys and passwords.

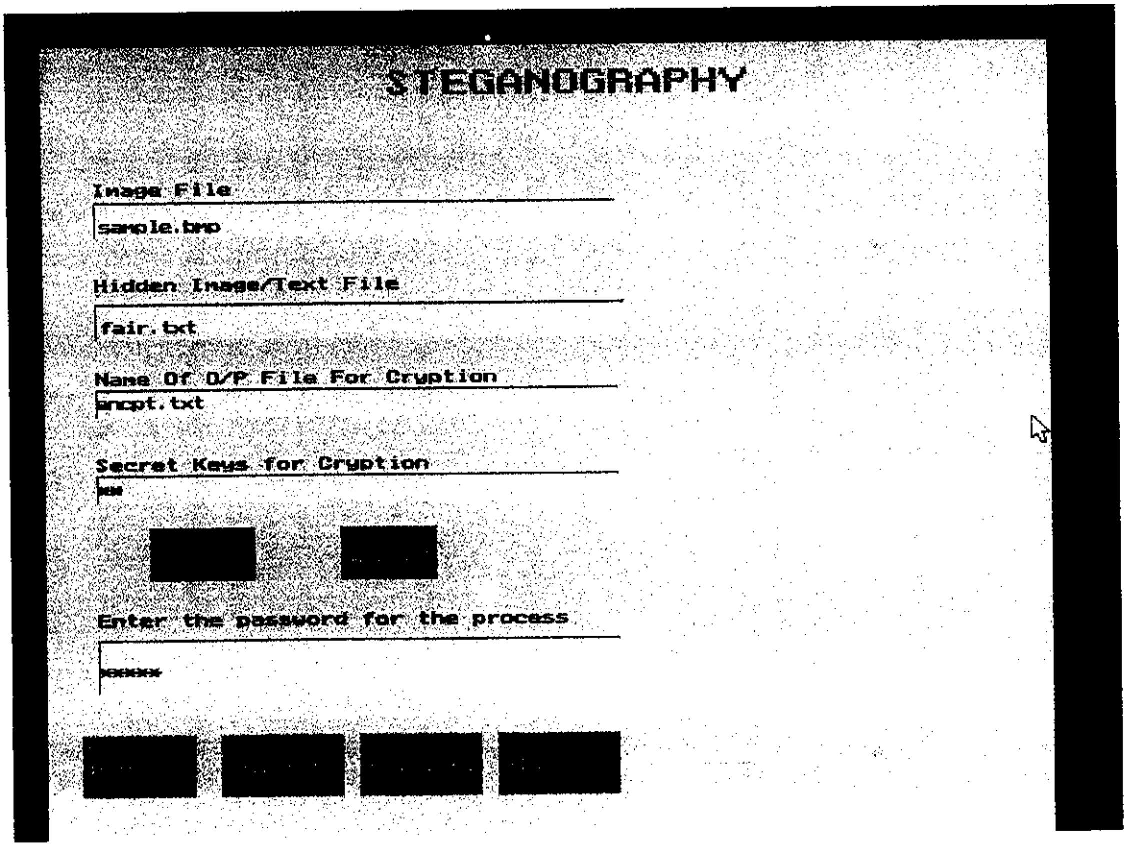
Mouse:

The user can make use of the mouse to click the options such as Hide, Retrieve, Cryption and Exit.

A.4 COMMAND SYNTAX AND SYSTEM OPTIONS:

The organization loads the executable format of the software "Steganography" in a CD and that it is ready for the users to make use of it. The users are provided with a readme text file, to use the software. The users can load the software to their Personal Computer, and they can make use of it, in hiding and retrieving the secret information within images.

A.5 Sample Screens and Output:



STEGANOGRAPHY

Image File

sample.bmp

Hidden Image/Text File

fair.txt

Name Of O/P File For Cryption

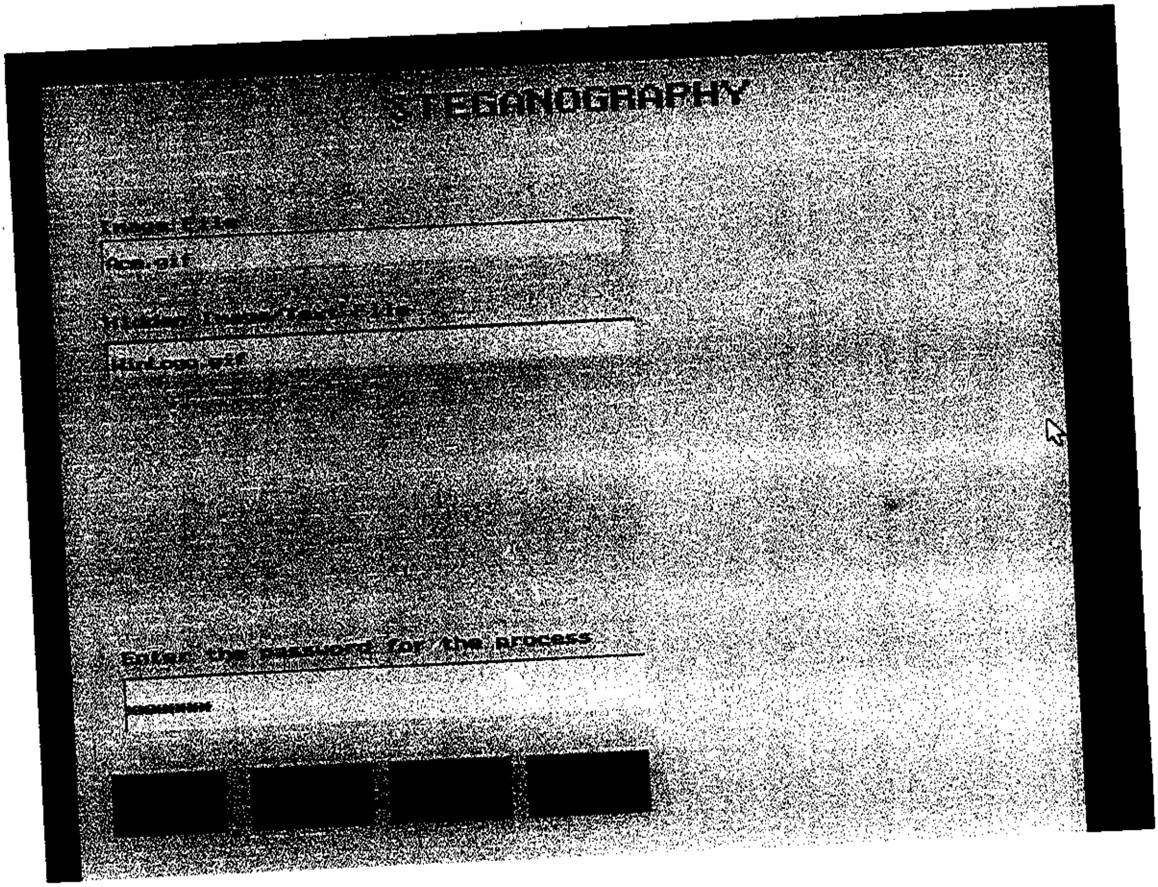
encpt.txt

Secret Keys for Cryption

**

Enter the password for the process

*****



STEGANOGRAPHY

Image File

Ace.gif

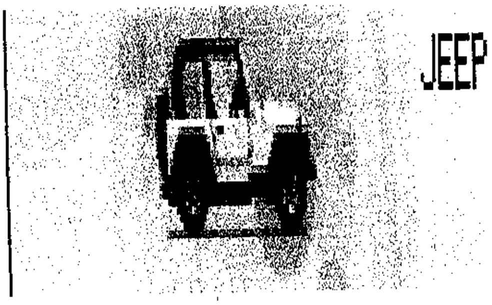
Hidden Image/Text File

HidLogo.gif

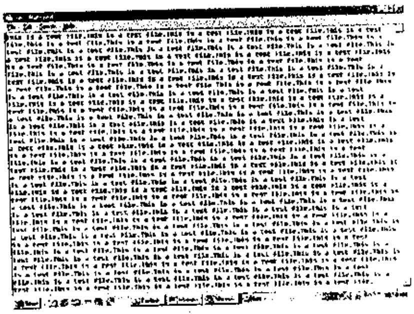
Enter the password for the process

****

Sample Input: Cover medium 8 bit bmp



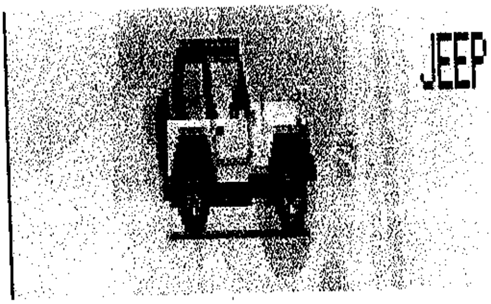
Sample Input: Hidden medium text file



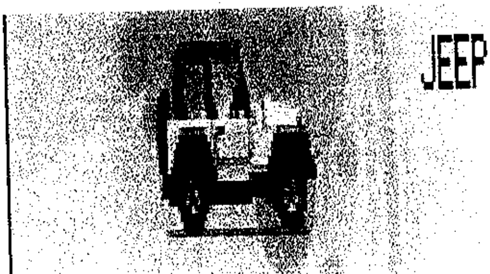
```
This is a test file.This is a test file.This is a test file.This is a test file.This is a test
This is a test file.This is a test file.This is a test file.This is a test file.This is a
file.This is a test file.This is a test file.This is a test file.This is a test file.This is
test file.This is a test file.This is a test file.This is a test file.This is a test file.This
a test file.This is a test file.This is a test file.This is a test file.This is a test
is a test file.This is a test file.This is a test file.This is a test file.This is a
file.This is a test file.This is a test file.This is a test file.This is a test file.This is
test file.This is a test file.This is a test file.This is a test file.This is a test file.This
a test file.This is a test file.This is a test file.This is a test file.This is a test
is a test file.This is a test file.This is a test file.This is a test file.This is a
file.This is a test file.This is a test file.This is a test file.This is a test file.This is
test file.This is a test file.This is a test file.This is a test file.This is a test file.This
a test file.This is a test file.This is a test file.This is a test file.This is a test
is a test file.This is a test file.This is a test file.This is a test file.This is a
file.This is a test file.This is a test file.This is a test file.This is a test file.This is
test file.This is a test file.This is a test file.This is a test file.This is a test file.This
a test file.This is a test file.This is a test file.This is a test file.This is a test
is a test file.This is a test file.This is a test file.This is a test file.This is a
file.This is a test file.This is a test file.This is a test file.This is a test file.This is
test file.This is a test file.This is a test file.This is a test file.This is a test file.This
a test file.This is a test file.This is a test file.This is a test file.This is a test
is a test file.This is a test file.This is a test file.This is a test file.This is a
file.This is a test file.This is a test file.This is a test file.This is a test file.This is
test file.This is a test file.This is a test file.This is a test file.This is a test file.This
a test file.This is a test file.This is a test file.This is a test file.This is a test
is a test file.This is a test file.This is a test file.This is a test file.This is a
file.This is a test file.This is a test file.This is a test file.This is a test file.This is
test file.This is a test file.This is a test file.This is a test file.This is a test file.This
a test file.This is a test file.This is a test file.This is a test file.This is a test
is a test file.This is a test file.This is a test file.This is a test file.This is a
file.This is a test file.This is a test file.This is a test file.This is a test file.This is
test file.This is a test file.This is a test file.This is a test file.This is a test file.This
a test file.This is a test file.This is a test file.This is a test file.This is a test
is a test file.This is a test file.This is a test file.This is a test file.This is a
file.This is a test file.This is a test file.This is a test file.This is a test file.This is
test file.This is a test file.This is a test file.This is a test file.This is a test file.This
```

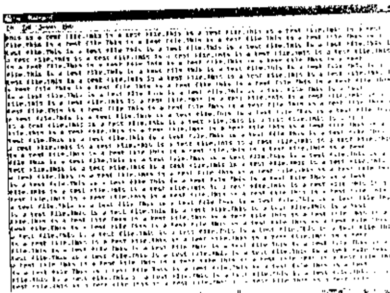
OUTPUT

Stego media



Output after retrieval process





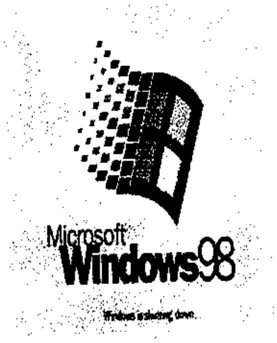
Sample Input: Cover medium 8 bit bmp

Microsoft
Windows98
Windows is shutting down.

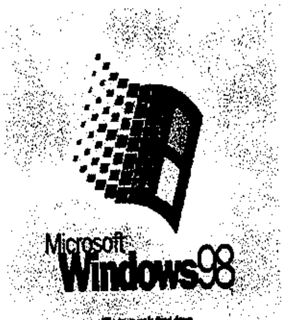
Sample Input: Hidden medium 8 bit bmp

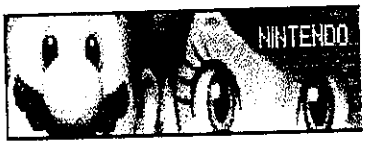
OUTPUT

Stego media

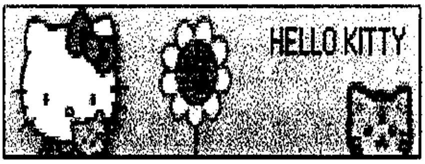


Output after retrieval process

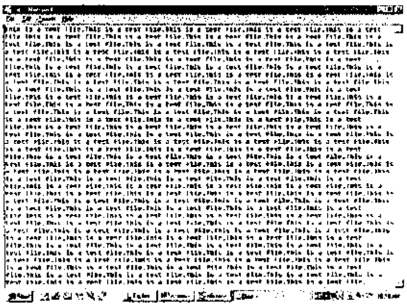




Sample Input: Cover medium 16 bit bmp



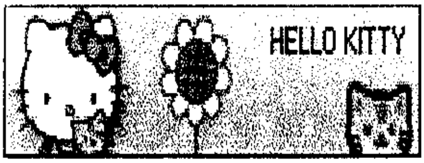
Sample Input: Hidden medium text file



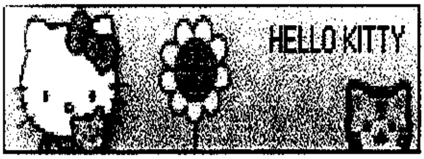
```
This is a test file.This is a test file.This is a test file.This is a test file.This is a test
file.This is a test file.This is a test file.This is a test file.This is a test file.This is a
test file.This is a test file.This is a test file.This is a test file.This is a test file.This is
a test file.This is a test file.This is a test file.This is a test file.This is a test file.This
is a test file.This is a test file.This is a test file.This is a test file.This is a test
file.This is a test file.This is a test file.This is a test file.This is a test file.This is a
test file.This is a test file.This is a test file.This is a test file.This is a test file.This is
a test file.This is a test file.This is a test file.This is a test file.This is a test file.This
is a test file.This is a test file.This is a test file.This is a test file.This is a test
file.This is a test file.This is a test file.This is a test file.This is a test file.This is a
test file.This is a test file.This is a test file.This is a test file.This is a test file.This is
a test file.This is a test file.This is a test file.This is a test file.This is a test file.This
is a test file.This is a test file.This is a test file.This is a test file.This is a test
file.This is a test file.This is a test file.This is a test file.This is a test file.This is a
test file.This is a test file.This is a test file.This is a test file.This is a test file.This is
a test file.This is a test file.This is a test file.This is a test file.This is a test file.This
is a test file.This is a test file.This is a test file.This is a test file.This is a test
file.This is a test file.This is a test file.This is a test file.This is a test file.This is a
test file.This is a test file.This is a test file.This is a test file.This is a test file.This is
a test file.This is a test file.This is a test file.This is a test file.This is a test file.This
is a test file.This is a test file.This is a test file.This is a test file.This is a test
file.This is a test file.This is a test file.This is a test file.This is a test file.This is a
test file.This is a test file.This is a test file.This is a test file.This is a test file.This is
a test file.This is a test file.This is a test file.This is a test file.This is a test file.This
is a test file.This is a test file.This is a test file.This is a test file.This is a test
file.This is a test file.This is a test file.This is a test file.This is a test file.This is a
test file.This is a test file.This is a test file.This is a test file.This is a test file.This is
a test file.This is a test file.This is a test file.This is a test file.This is a test file.This
is a test file.This is a test file.This is a test file.This is a test file.This is a test
file.This is a test file.This is a test file.This is a test file.This is a test file.This is a
test file.This is a test file.This is a test file.This is a test file.This is a test file.This is
a test file.This is a test file.This is a test file.This is a test file.This is a test file.This
is a test file.This is a test file.This is a test file.This is a test file.This is a test
file.This is a test file.This is a test file.This is a test file.This is a test file.This is a
test file.This is a test file.This is a test file.This is a test file.This is a test file.
```

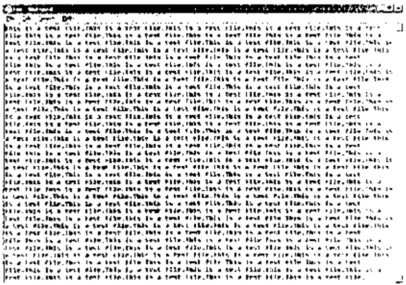
OUTPUT

Stego media



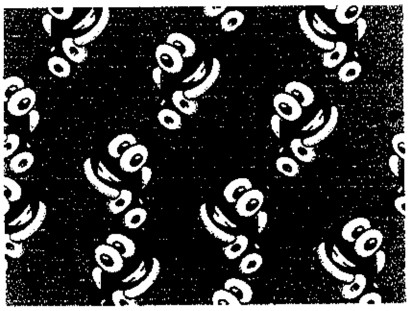
Output after retrieval process





```
This is a test file.This is a test file.This is a test file.This is a test file.This is a test
file.This is a test file.This is a test file.This is a test file.This is a test file.This is a
test file.This is a test file.This is a test file.This is a test file.This is a test file.This is
a test file.This is a test file.This is a test file.This is a test file.This is a test file.This
is a test file.This is a test file.This is a test file.This is a test file.This is a test
file.This is a test file.This is a test file.This is a test file.This is a test file.This is a
test file.This is a test file.This is a test file.This is a test file.This is a test file.This is
a test file.This is a test file.This is a test file.This is a test file.This is a test file.This
is a test file.This is a test file.This is a test file.This is a test file.This is a test
file.This is a test file.This is a test file.This is a test file.This is a test file.This is a
test file.This is a test file.This is a test file.This is a test file.This is a test file.This is
a test file.This is a test file.This is a test file.This is a test file.This is a test file.This
is a test file.This is a test file.This is a test file.This is a test file.This is a test
file.This is a test file.This is a test file.This is a test file.This is a test file.This is a
test file.This is a test file.This is a test file.This is a test file.This is a test file.This is
a test file.This is a test file.This is a test file.This is a test file.This is a test file.This
is a test file.This is a test file.This is a test file.This is a test file.This is a test
file.This is a test file.This is a test file.This is a test file.This is a test file.This is a
test file.This is a test file.This is a test file.This is a test file.This is a test file.This is
a test file.This is a test file.This is a test file.This is a test file.This is a test file.This
is a test file.This is a test file.This is a test file.This is a test file.This is a test
file.This is a test file.This is a test file.This is a test file.This is a test file.This is a
test file.This is a test file.This is a test file.This is a test file.This is a test file.This is
a test file.This is a test file.This is a test file.This is a test file.This is a test file.This
is a test file.This is a test file.This is a test file.This is a test file.This is a test
file.This is a test file.This is a test file.This is a test file.This is a test file.This is a
test file.This is a test file.This is a test file.This is a test file.This is a test file.This is
a test file.This is a test file.This is a test file.This is a test file.This is a test file.This
is a test file.This is a test file.This is a test file.This is a test file.This is a test
file.This is a test file.This is a test file.This is a test file.This is a test file.This is a
test file.This is a test file.This is a test file.This is a test file.This is a test file.This is
a test file.This is a test file.This is a test file.This is a test file.This is a test file.This
is a test file.This is a test file.This is a test file.This is a test file.This is a test
file.This is a test file.This is a test file.This is a test file.This is a test file.This is a
test file.This is a test file.This is a test file.This is a test file.This is a test file.
```

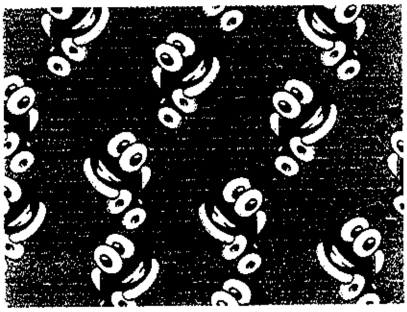
Sample Input: Cover medium 16 bit bmp



Sample Input: Hidden medium 16 bit bmp

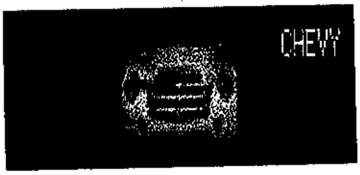
OUTPUT

Stego media

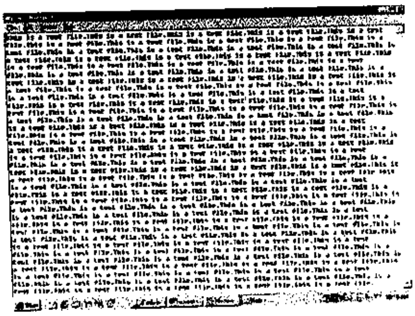


Output after retrieval process

Sample Input: Cover medium 24 bit bmp

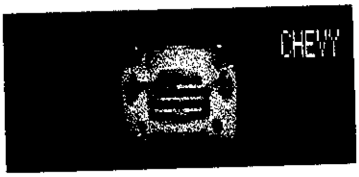


Sample Input: Hidden medium text file

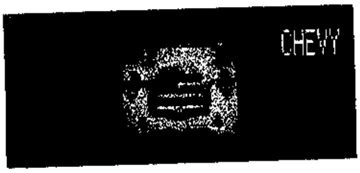


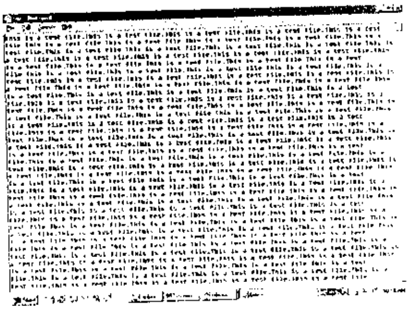
OUTPUT

Stego media

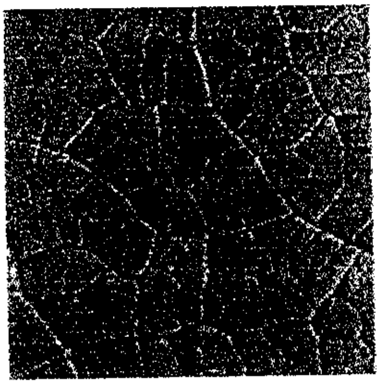


Output after retrieval process





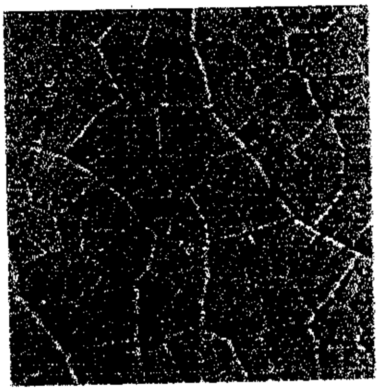
Sample Input: Cover medium 24 bit bmp



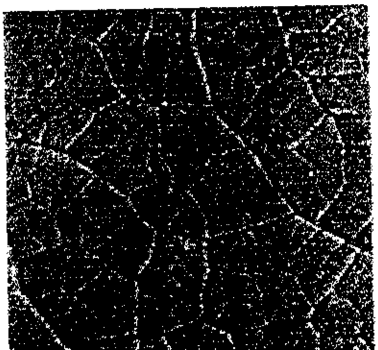
Sample Input: Hidden medium 24 bit bmp

OUTPUT

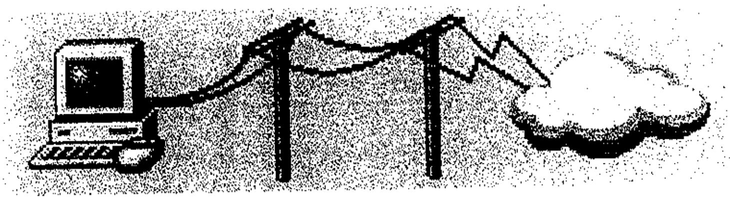
Stego media



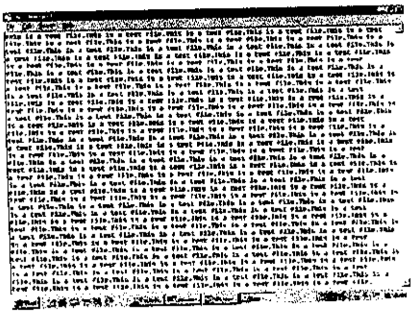
Output after retrieval process



Sample Input: Cover medium 8 bit gif



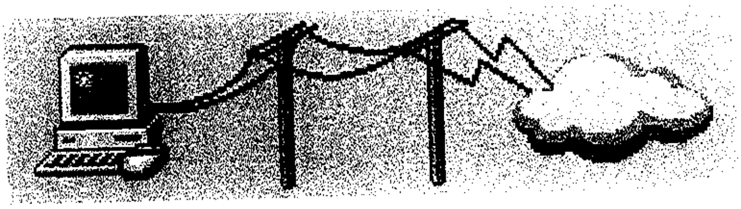
Sample Input: Hidden medium text file



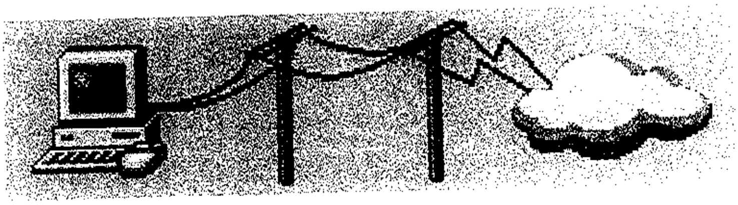
```
This is a test file.This is a test file.This is a test file.This is a test file.This is a test
file.This is a test file.This is a test file.This is a test file.This is a test file.This is a
test file.This is a test file.This is a test file.This is a test file.This is a test file.This is
a test file.This is a test file.This is a test file.This is a test file.This is a test file.This
is a test file.This is a test file.This is a test file.This is a test file.This is a test
file.This is a test file.This is a test file.This is a test file.This is a test file.This is a
test file.This is a test file.This is a test file.This is a test file.This is a test file.This
a test file.This is a test file.This is a test file.This is a test file.This is a test
is a test file.This is a test file.This is a test file.This is a test file.This is a
file.This is a test file.This is a test file.This is a test file.This is a test file.This is
test file.This is a test file.This is a test file.This is a test file.This is a test file.This
a test file.This is a test file.This is a test file.This is a test file.This is a test
is a test file.This is a test file.This is a test file.This is a test file.This is a
file.This is a test file.This is a test file.This is a test file.This is a test file.This is
test file.This is a test file.This is a test file.This is a test file.This is a test file.This
a test file.This is a test file.This is a test file.This is a test file.This is a test
is a test file.This is a test file.This is a test file.This is a test file.This is a
file.This is a test file.This is a test file.This is a test file.This is a test file.This is
test file.This is a test file.This is a test file.This is a test file.This is a test file.This
a test file.This is a test file.This is a test file.This is a test file.This is a test
is a test file.This is a test file.This is a test file.This is a test file.This is a
file.This is a test file.This is a test file.This is a test file.This is a test file.This is
test file.This is a test file.This is a test file.This is a test file.This is a test file.This
a test file.This is a test file.This is a test file.This is a test file.This is a test
is a test file.This is a test file.This is a test file.This is a test file.This is a
file.This is a test file.This is a test file.This is a test file.This is a test file.This is
test file.This is a test file.This is a test file.This is a test file.This is a test file.This
a test file.This is a test file.This is a test file.This is a test file.This is a test
is a test file.This is a test file.This is a test file.This is a test file.This is a
file.This is a test file.This is a test file.This is a test file.This is a test file.This is
test file.This is a test file.This is a test file.This is a test file.This is a test file.This
a test file.This is a test file.This is a test file.This is a test file.This is a test
is a test file.This is a test file.This is a test file.This is a test file.This is a
file.This is a test file.This is a test file.This is a test file.This is a test file.This is
test file.This is a test file.This is a test file.This is a test file.This is a test file.
```

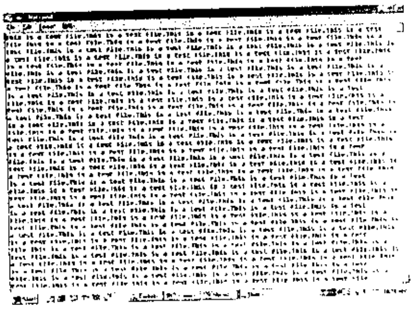
OUTPUT

Stego media



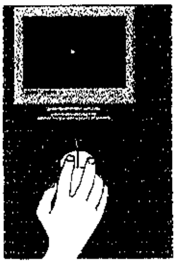
Output after retrieval process





```
This is a test file.This is a test file.This is a test file.This is a test file.This is a test
file.This is a test file.This is a test file.This is a test file.This is a test file.This is a
test file.This is a test file.This is a test file.This is a test file.This is a test file.This is
a test file.This is a test file.This is a test file.This is a test file.This is a test file.This
is a test file.This is a test file.This is a test file.This is a test file.This is a test
file.This is a test file.This is a test file.This is a test file.This is a test file.This is a
test file.This is a test file.This is a test file.This is a test file.This is a test file.This
a test file.This is a test file.This is a test file.This is a test file.This is a test
is a test file.This is a test file.This is a test file.This is a test file.This is a
file.This is a test file.This is a test file.This is a test file.This is a test file.This is
test file.This is a test file.This is a test file.This is a test file.This is a test file.This
a test file.This is a test file.This is a test file.This is a test file.This is a test
is a test file.This is a test file.This is a test file.This is a test file.This is a
file.This is a test file.This is a test file.This is a test file.This is a test file.This is
test file.This is a test file.This is a test file.This is a test file.This is a test file.This
a test file.This is a test file.This is a test file.This is a test file.This is a test
is a test file.This is a test file.This is a test file.This is a test file.This is a
file.This is a test file.This is a test file.This is a test file.This is a test file.This is
test file.This is a test file.This is a test file.This is a test file.This is a test file.This
a test file.This is a test file.This is a test file.This is a test file.This is a test
is a test file.This is a test file.This is a test file.This is a test file.This is a
file.This is a test file.This is a test file.This is a test file.This is a test file.This is
test file.This is a test file.This is a test file.This is a test file.This is a test file.This
a test file.This is a test file.This is a test file.This is a test file.This is a test
is a test file.This is a test file.This is a test file.This is a test file.This is a
file.This is a test file.This is a test file.This is a test file.This is a test file.This is
test file.This is a test file.This is a test file.This is a test file.This is a test file.This
a test file.This is a test file.This is a test file.This is a test file.This is a test
is a test file.This is a test file.This is a test file.This is a test file.This is a
file.This is a test file.This is a test file.This is a test file.This is a test file.This is
test file.This is a test file.This is a test file.This is a test file.This is a test file.This
a test file.This is a test file.This is a test file.This is a test file.This is a test
is a test file.This is a test file.This is a test file.This is a test file.This is a
file.This is a test file.This is a test file.This is a test file.This is a test file.This is
test file.This is a test file.This is a test file.This is a test file.This is a test file.
```

Sample Input: Cover medium 8 bit gif

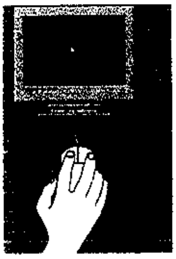


Sample Input: Hidden medium 8 bit gif

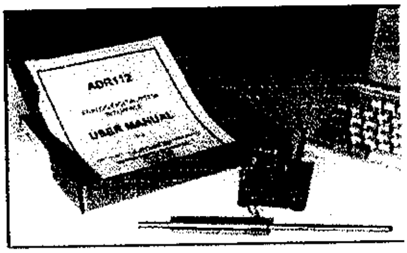
OUTPUT

Stego media

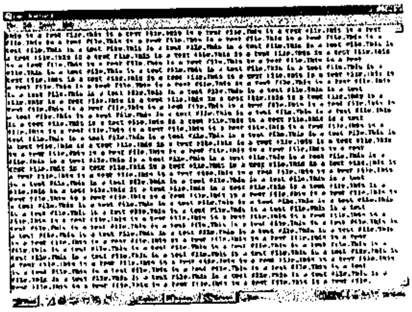
Output after retrieval process



Sample Input: Cover medium 24 bit JPEG

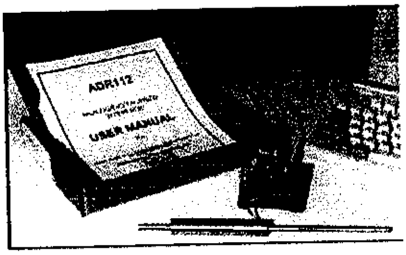


Sample Input: Hidden medium text file

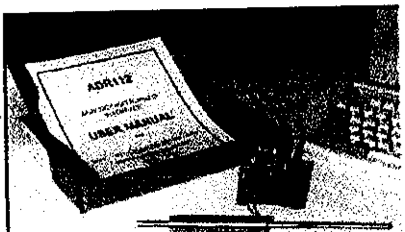


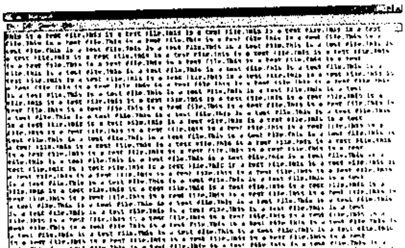
OUTPUT

Stego media



Output after retrieval process



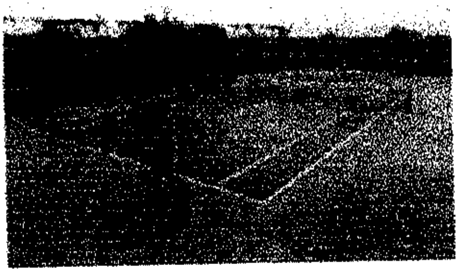


Sample Input: Cover medium 24 bit JPEG

Sample Input: Hidden medium 24 bit JPEG

OUTPUT

Stego media



Output after retrieval process