**1.INTRODUCTION:**

* 1. **Concept**

The term steganography is derived from the Greek words “Steganos” (covered) and “Graphia” (writing).The intension of Steganography is to provide the secret transmission of data.

Steganalysis provides a way of detecting the presence of hidden information.

**1.2 History**

Steganography methods have been used for centuries. In ancient Greek times, messengers tattooed messages on their shaved heads and the messages remain invisible when their hair grows. Wax tables were used as cover source. Message to be hidden was written on the wood and was covered with new wax layer. During Second World War, milk, fruit juices, vinegar were used for writing secret messages. Invisible inks Carrier medium were used to hide information in 20th century .During 1990’s secret messages were hidden into some digital files. Government, industries and terrorist organization use steganography for hiding secret data.

**1.3** **Image Steganography**

Image steganography is defined as the covert embedding of data into digital pictures. Though steganography hides information in any one of the digital Medias, digital images are the most popular as carrier due to their frequency usage on the internet. Since the size of the image file is large, it can conceal large amount of information. HVS (Human Visual System) cannot differentiate the normal image and the image with hidden data. In addition with that digital images includes large amount of redundant bits, images became the most popular cover objects for steganography. Hence this research uses image as cover file.

Different image formats such as JPEG, BMP, TIFF, PNG or GIF files can be used as cover objects. A bitmap or BMP format is a simple image file format. Data is easy to manipulate, since it is uncompressed. But the uncompressed data leads to larger file size than the compressed image.JPEG(Joint Photographic Expert Group) is the most commonly used image file format. It uses lossy compression technique; the quality of the image is excellent. The size of the file is also smaller.TIFF format uses lossless compression. The file is reduced without affecting the image quality. GIF (Graphics Interchange format) has color palette to provide an indexed colors image. It uses lossless compression. Since it can store only 256 different colors it is not suitable for representing complex photography with continuous tones, PNG (Portable Network Graphics) file format provides better colors support, best compression, and gamma correction in brightness control and image transparency. PNG format can be used as an alternative to GIF to represent web images.

**1.4 Types of images**

Digital image is represented as a set of picture element called pixel. They are organized as two dimensional arrays. Digital images can be classified according to the number of bits per pixel since the number of distinct colors of a digital image depends on number bits per pixel (bpp). There are three common types of images: a) Binary image: In this type, one bit is allocated for each pixel. The value of a bit is represented as either 1 or 0. Each pixels of a binary image should be represented as any one of two colors (black and white). Binary image is also called as bi-level image.

b) Gray scale image: A digital image, in which the colors are represented as shades of grey, is known as grey scale image. The darkest possible shade is black, where as the highest shade is white. Each pixel is represented using eight bits. Hence, it can create 256 different shades of grey.

c) RGB (true) color image: The color of each pixel is determined by the combination of red, green and blue intensities. Each pixel is represented using 24 bits, where red, green and blue components are 8 bits each. Hence, 16.7 million possible distinct colors may be represented. There are two types of image file compression algorithms: lossless and lossy.

“Lossless compression” algorithms reduce file size while preserving a perfect copy of the original uncompressed image. Lossless compression generally, but not always, results in larger files than lossy compression. Lossless compression should be used to avoid accumulating stages of re-compression when editing images.

“Lossy compression” algorithms preserve a representation of the original uncompressed image that may appear to be a perfect copy, but it is not a perfect copy. Often lossy compression is able to achieve smaller file sizes than lossless compression. Most lossy compression algorithms allow for variable compression that trades image quality for file size.

Steganography and Steganalysis have received a lot of attention around the world in the past few years. Some are interested in securing their communications through hiding the very own fact that they are exchanging information. On the other hand, others are interested in detecting the existence of these communications – possibly because they might be related to illegal activities.

**1.5 Terminology**

According to the general model of Information Hiding: embedded data is the message we want to send secretly. Often, we hide the embedded data in an innocuous medium, called cover message. There are many kinds of cover messages such as cover text, when we use text to hide a message; or cover image, when we use an image to hide a message. The embedding process produces a stego object which contains the hidden message. We can use a stego key to control the embedding process, so we can also restrict detection and/or recovery of the embedded data to other parties with the appropriate permissions to access this data. Figure 1 shows the process of hiding a message in an image. First we choose the data we want to hide. Further, we use a selected key to hide the message in a previously selected cover image which produces the stego image. Covert channels consist of the use of a secret and secure channel for communication purposes (eg Military Covert Channels).

Steganography is the art, and science, of hiding the information to avoid its detection.

**2.LITERATURE SURVEY:**

**2.1 C# Language and .Net Platform**

The entire Application has been developed using c#. .Net has been chosen as the platform because of its feature rich nature. C# is a powerful Programming language with excellent features in .Net platform like: Interoperability,security, and Simplified deployment.

The .Net Platform provides robust end-to-end solutions as a trusted standard for embedded applications. So c# and .Net was a natural choice for development process.

**Characteristics of C# Progromming Language:-**

* Modern and Object-oriented
* Simple and Flexible
* Type Safety
* Automatic Memory Management
* Versioning Control
* Cross Platform Inter-operability

**.Net Features : -**

* Common Runtime Engine
* Inter-operability
* Complete and Total Language integration
* Base class Libraries
* Simplified Deployment

**Visual Studio 2010**

**Microsoft Visual Studio 2010** is a set of freeware integrated development environments(IDE) developed by Microsoft as a lightweight version of Microsoft Visual Studio. Express editions were conceived beginning with Visual Studio 2005. The idea of Visual Express editions is to provide streamlined, easy-to-use and easy-to-learn IDEs for hobbyists and students, rather than professional software developers.

Visual C# 2010 is an easy-to-use, free, lightweight, integrated development environment (IDE) designed for novice developers, students and hobbyists to create applications and (when combined with the XNA Game Studio) video games for Windows, Xbox and 360 and Zune.It can build console,Windows Forms and Windows Presentation Foundation applications as well as class libraries.

**2.2 Proposed System**

The Proposed system is the Steganography Testing Tool which is an application developed to Hide information within images and retrieve it. Steganographic tools are generally used to store Confidential Textual Information Within images and also securely transfer them across the internet. The Receiver can decrypt the info and thereby gets to know the confidential information from the sender. Password protected Encryption can be used for such Stegoimages.

**2.3 Advantages of Steganography:**

* Normal Person Doesn’t know Hidden information
* Enahancing Security and prevents Hacking
* Easy to use and Speedy Technique
* Flexibility: Large Number of Words can be added
* Technique Can be applied in a variety of fields like Medicine,agriculture,Technology etc..

**2.4 Problem Statement and objective**

This project is developed for hiding information in an image file. The scope of the project is implementation of steganography tools for hiding information includes any type of information file and image files( especially jpeg,png,bmp images) & the path where the user wants to save image and extruded file.

We can able to use information hiding techniques that may be used to exchange steganograms in telecommunication networks .This is also called as network Steganography.

Steganalysis is a relatively new approach “The process of detecting steganography by looking at variances between bit patterns and unusually large file sizes.

The goal of steganalysis is to identify suspected information streams,determine whether or not they have hidden messages encoded into them,and,if possible, recover the hidden information.

**3.REQUIREMENT SPECIFICATION**

**3.1 Software requirements:**

* .NET Framework 4.5.1
* C# Language
* Windows 7 Operating System

**3.2 Hardware requirements:**

* Processor: Preferably 1.0 GHz or greater..
* RAM: 512 MB or greater..

**4. PROJECT DESIGN AND IMPLEMENTATION:-**

**4.1 DESIGN TOOL**

**4.1.1 : Microsoft Visual Studio 2010** is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs for Microsoft Windows superfamily of operating systems, as well as web sites, web applications and web services. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.

Visual Studio includes a code editor supporting IntelliSense as well as code refactoring. The integrated debugger works both as a source-level debugger and a machine-level debugger. Other built-in tools include a forms designer for building [GUI](http://en.wikipedia.org/wiki/GUI) applications, web designer, class designer, and database schema designer. It accepts plug-ins that enhance the functionality at almost every level—including adding support for source-control systems (like Subversion) and adding new toolsets like editors and visual designers for domain-specific languages or toolsets for other aspects of the software development lifecycle(like the Team Foundation Server client: Team Explorer).

Visual Studio supports different programming languages and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include C, C++and C++/CLI (via Visual C++), VB.NET (via Visual Basic .NET), C# (via Visual C#), and F# (as of Visual Studio 2010). Support for other languages such as M, Python, and Ruby among others is available via language services installed separately. It also supports [XML](http://en.wikipedia.org/wiki/XML)/[XSLT](http://en.wikipedia.org/wiki/XSLT), [HTML](http://en.wikipedia.org/wiki/HTML)/[XHTML](http://en.wikipedia.org/wiki/XHTML), [JavaScript](http://en.wikipedia.org/wiki/JavaScript) and [CSS](http://en.wikipedia.org/wiki/Cascading_Style_Sheets).

In this project we are trying to implement the concepts that we have learnt in C# and .Net. We have used some basic functions of WidowsForm and we have created Steganography Application.

We have used Visual Basic components to build the GUI which is helpful for the user to give inputs in fast and efficient manner.

In this model there are mainly six types of operations, namely insertion of image, insertion of text, Password Protection, Encryption of Secret Message, Save the Image(Stego-Image) in Required Location and Decrypt the Stego-Image.

We have used the GUI components to do the operations. We have created the main GUI that consists of the separate buttons for each operation and a picture box to load the image. When a user clicks a button, the respective sub-GUI will come,the corresponding code of the function it must serve has been written and it is helpful for the user to do the corresponding operation.

**4.1.2 System Namespaces:**

1)System.Drawing:

The System.Drawing  namespace provides graphical capabilities.  
With its tools,we manipulate and create visual effects. We handle images and primitive shapes. The Image and Font types are among the most important.This console program opens an image with the Image.FromFile method. It then prints the size of the image to the console. The Image type makes working with image data, such as PNG and JPG, easier.

2)System.Windows.Forms: **Windows Forms** (WinForms) is the name given to the graphical application programming interface (API) included as a part ofMicrosoft .NET Framework, providing access to native Microsoft Windows interface elements by wrapping the extant Windows API in managed code. While it is seen as a replacement for the earlier and more complex C++ based Microsoft Foundation Class Library, it does not offer a paradigm comparable to Model–View–Controller. Some after-market and third party libraries have been created to provide this functionality. A Windows *Form*sapplication is an event-driven application supported by Microsoft's .NET Framework. Unlike a batch program, it spends most of its time simply waiting for the user to do something, such as fill in a text box or click a button.

3) System.Drawing.Imaging: An abstract base class that provides functionality for the Bitmap and Metafile descended classes.

## [Inheritance Hierarchy](javascript:void(0)) :

[System.Object](http://msdn.microsoft.com/en-us/library/vstudio/system.object)   
[System.MarshalByRefObject](http://msdn.microsoft.com/en-us/library/vstudio/system.marshalbyrefobject)  
System.Drawing.Image  
[System.Drawing.Bitmap](http://msdn.microsoft.com/en-us/library/vstudio/system.drawing.bitmap)  
[System.Drawing.Imaging.Metafile](http://msdn.microsoft.com/en-us/library/vstudio/system.drawing.imaging.metafile)

**Namespace:**  [System.Drawing](http://msdn.microsoft.com/en-us/library/vstudio/system.drawing)  
**Assembly:**  System.Drawing (in System.Drawing.dll)

4) System.IO: The System.IO namespace contains types that allow reading and writing to files and data streams, and types that provide basic file and directory support.

System.IO provides all the necessary classes, methods, and properties for manipulating directories and files. Table 1 elaborates the main classes under this namespace.

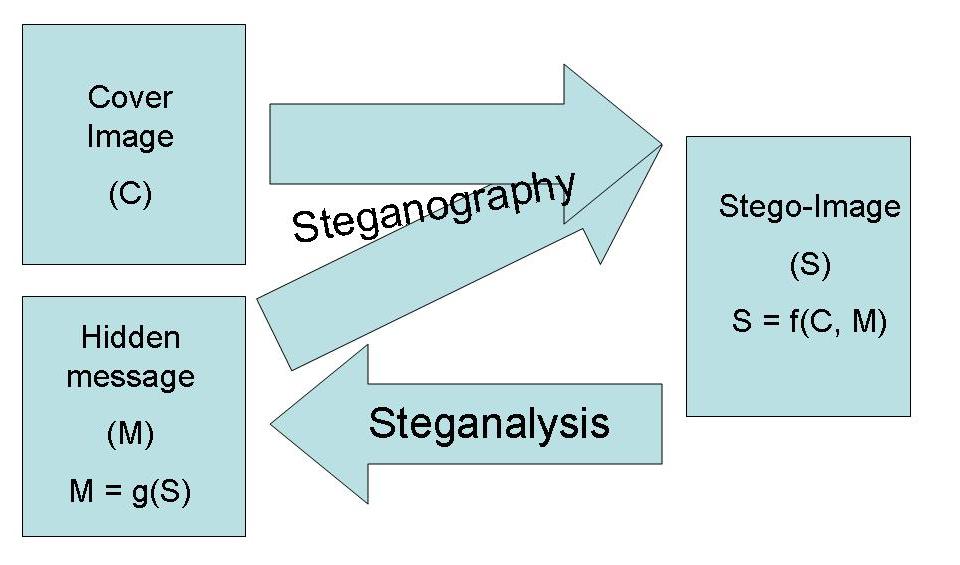
|  |  |
| --- | --- |
| **Class** | **Purpose/Use** |
| Binary Reader and Writer | Read and write primitive data types |
| Directory, File, DirectoryInfo, and FileInfo | Create, delete, and move files and directories. Get specific information about the files by making use of the properties defined in these classes. |
| FileStream | Access the files in a random fashion |
| MemoryStream | Access data stored in memory |
| StreamWriter and StreamReader | Read and write textual information |
| StringReader and StringWriter | Read and write textual Information from a string buffer |

5)System.Security.Cryptography: The System.Security.Cryptography namespace provides cryptographic services, including secure encoding and decoding of data, as well as many other operations, such as hashing, random number generation, and message authentication.

|  |  |
| --- | --- |
| **Class** | **Description** |
| [Aes](http://msdn.microsoft.com/en-us/library/system.security.cryptography.aes.aspx) | Represents the abstract base class from which all implementations of the Advanced Encryption Standard (AES) must inherit. |
| [AesCryptoServiceProvider](http://msdn.microsoft.com/en-us/library/system.security.cryptography.aescryptoserviceprovider.aspx) | Performs symmetric encryption and decryption using the Cryptographic Application Programming Interfaces (CAPI) implementation of the Advanced Encryption Standard (AES) algorithm. |
| [CngAlgorithm](http://msdn.microsoft.com/en-us/library/system.security.cryptography.cngalgorithm.aspx) | Encapsulates the name of an encryption algorithm. |
| [DES](http://msdn.microsoft.com/en-us/library/system.security.cryptography.des.aspx) | Represents the base class for the Data Encryption Standard (DES) algorithm from which all [DES](http://msdn.microsoft.com/en-us/library/system.security.cryptography.des.aspx) implementations must derive. |
| [DESCryptoServiceProvider](http://msdn.microsoft.com/en-us/library/system.security.cryptography.descryptoserviceprovider.aspx) | Defines a wrapper object to access the cryptographic service provider (CSP) version of the Data Encryption Standard ([DES](http://msdn.microsoft.com/en-us/library/system.security.cryptography.des.aspx)) algorithm. This class cannot be inherited. |
| [ToBase64Transform](http://msdn.microsoft.com/en-us/library/system.security.cryptography.tobase64transform.aspx) | Converts a [CryptoStream](http://msdn.microsoft.com/en-us/library/system.security.cryptography.cryptostream.aspx) to base 64. |
| [ProtectedData](http://msdn.microsoft.com/en-us/library/system.security.cryptography.protecteddata.aspx) | Provides methods for encrypting and decrypting data. This class cannot be inherited. |

4.2 IMPLEMENTATION:

4.2.1 )Base Concept:



This project simulates the working of Steganography. The Figure shows the movement as how the Steganography and Steganalysis works. It has a wide range of user interaction as well.

C:\Users\tnr\AppData\Local\Temp\ksohtml\wps_clip_image-7269.png

C:\Users\tnr\AppData\Local\Temp\ksohtml\wps_clip_image-7315.png

C:\Users\tnr\AppData\Local\Temp\ksohtml\wps_clip_image-7374.png

C:\Users\tnr\AppData\Local\Temp\ksohtml\wps_clip_image-7403.png

C:\Users\tnr\AppData\Local\Temp\ksohtml\wps_clip_image-7488.png

C:\Users\tnr\AppData\Local\Temp\ksohtml\wps_clip_image-7534.png

C:\Users\tnr\AppData\Local\Temp\ksohtml\wps_clip_image-7579.png

C:\Users\tnr\AppData\Local\Temp\ksohtml\wps_clip_image-7615.png

**Fig : Steganography Flow Chart**

**Usages**: Click the button and choose to select operation and select the views and exit.  
**First button :** To select the new Steganographic Approach presented in the Journal.

**Second button:**To Select LSB steganographic technique.  
**OpenFilePath Button :**To Load the Image onto which encryption or decryption is performed.

**TextField:**To list the text message to Encrypt or lists the decrypted Text.

**Password Protection Button :**To protect Stego-Image by providing a password.

**Encrypt Button :** To Encrypt the secret Message onto the Image.

**Decrypt Button :** To Decrypt the secret Message onto the Textfield.

**Save Button :** The Location where we want to save the image.

**4.2.2)A New Approach to Hide Text in Images Using Steganography**:

**Proposed Algorithm** The algorithm that we have proposed in this system is basically an extension of the original LSB which is quite vulnerable. Instead of hiding the data in least significant bits of the RGB components of a pixel, we in this algorithm, would be hiding data as shown below: - Let the data to be hidden is word “ABC” ASCII code of A= 65 and corresponding binary is 01000001. ASCII code of B= 66 and corresponding binary is 01000010. ASCII code of C= 67 and corresponding binary is 01000011.

Let the first pixel‟s RGB component be: -

****

Red component is replaced with binary of 65 i.e. A.

****

Let the second pixel‟s RGB component be: -

****

Green component of second pixel is replaced with binary of 66 i.e. B.

****

Let the third pixel‟s RGB component be: -

****

Blue component of third pixel is replaced with binary of 67 i.e. C.

****

And the process continues. The resulting stego image that we are obtaining after the algorithm completes its execution, is distorted and is easy to detect, that some kind of alteration has been done to the image. So, to enhance the security of the secret message we would be covering the resulting stego image with a new cover image, this is the first level of security. By just looking at the resulting image no one would be able to predict that something is hidden inside it. The new cover image can be the same or different than the original. In order to increase the storage capacity of the image, a compression algorithm has been used, we know that each component of an RGB pixel is represented with 8 bits. So, the maximum compression would be 8 bits per pixel and minimum would be 1 bit per pixel. The proposed steganographic algorithm comprises of two embedding techniques they are data hiding technique and data retrieving technique. Data hiding technique as the name suggests is used to hide secret message and key in the cover image, while data retrieving technique is used to retrieve the key and the hidden secret message from the stego image. Therefore data is protected in image without revealing to unauthorized party.

***A. Proposed embedding technique.*** Inputs: - Text file, cover image 1, cover image 2 and secret key. Output: - Stego image. **Begin**

1. Select a text file, convert it into binary form and calculate the number of bits in it.

2. Select a carrier image (cover image 1) for hiding purpose, find the number of pixels, convert it into RGB image and calls the compression function.

3. If bits calculated are compatible with the image resolution, then

**Start sub iteration 1** Replace red component of the first pixel with first character. Replace green component of the second pixel with second character. Replace blue component of the third pixel with third character. And repeat iterations until pixels get exhaust.

**Stop sub iteration 1**

Else **Repeat sub iteration 1** Finds necessary compression ratio and perform sub iteration 2.

**Sub iteration 2** Replace necessary bits as defined by the compression ratio in immediate component of each pixel. Store the information about bits embedded in a binary address file.

**Stop sub iteration2**

4. Provide a security key as encryption completes.

5. Select 2nd cover image to hide the distorted stego image.

**End**

***B. Proposed extraction technique.*** Input**: -** Stego image and secret key. Output: - Secret text file.

**Begin** 1. Browse the stego image. 2. Choose the folder in which you want to extract the hidden text file. 3. Provide necessary security key. 4. Convert the binary file into human readable form. **End**

The main focuses of this proposed steganographic technique is to hide text files in images, compresses the text files so as to increase the overall storage capacity, applying a secret key on the resulting stego image and transferring the secret message without any vulnerability and threat.

****

**Figure 2: General Layout of Proposed System**

The system that we have developed in VB is able to maintain the accuracy, confidentiality of the data. The system is able to hide the text files in images using a secret key and also is able to retrieve the data back from the stego image.

4.2.3)LSB ALGORITHM: LSB algorithm:

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

Step1:

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

Step2:

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

Step3:

Let n LSBs be substituted in each pixel.

Step4:

Let d= decimal value of the pixel after the substitution.d1 = decimal value of last n bits of the pixel.d2 = decimal value of n bits hidden in that pixel.

Step5:

If (d1~d2)<=(2^n)/2then no adjustment is made in that pixel.

Else

Step6:

If(d1<d2)d = d – 2^n.If(d1>d2)d = d + 2^n.This ‘d’ is converted to binary and written back to pixel. This method of substitution is simple and easy to retrieve the data and the image quality better so that it provides good security.The encoder algorithm is as given below:

1: for i = 1, ..., len(msg) do

2: p = LSB(pixel of the image)3: if p != message bit then

4: pixel of the image = message bit

5: end if

6: end for The encoding process shows that the entire algorithm can be implemented by writing just a few lines of code. The algorithm works by taking the first pixel of the image and obtaining its LSB value (as per line 2 of the Algorithm). This is typically achieved by calculating the modulus 2 of the pixel value. This will return a 0 if the number is even, and a 1 if the number is odd, which effectively tells us the LSB value. We then compare this value with the message bit that we are trying to embed.If they are already the same, then we do nothing, but if they are different then were place the pixel value with the message bit. This process continues whilst there are still values in the message that need to be encoded The decoder algorithm is:

1: for i = 1, ..., len(image string) do

2:message string = LSB (pixel string of the image)

3: end for The decoding phase is even simpler. As the encoder replaced the LSBs of the pixel values in c in sequence, we already know the order that should be used to retrieve the data. Therefore all we need to do is calculate the modulus 2 of all the pixel values in the stegogramme, and we are able to reconstruct m as m0 .The above Algorithm shows the pseudo code of the decoding process.

\* Note that this time we run the loop for length of message instead of length of string. This is because the decoding process is completely separate from the encoding process and therefore has no means of knowing the length of the message. If a key were used, it would probably reveal this information, but instead we simply retrieve the LSB value of every pixel. When we convert this to ASCII, the message will be readable up to the point that the message was encoded, and will then appear as gibberish when we are reading the LSBs of the image data.

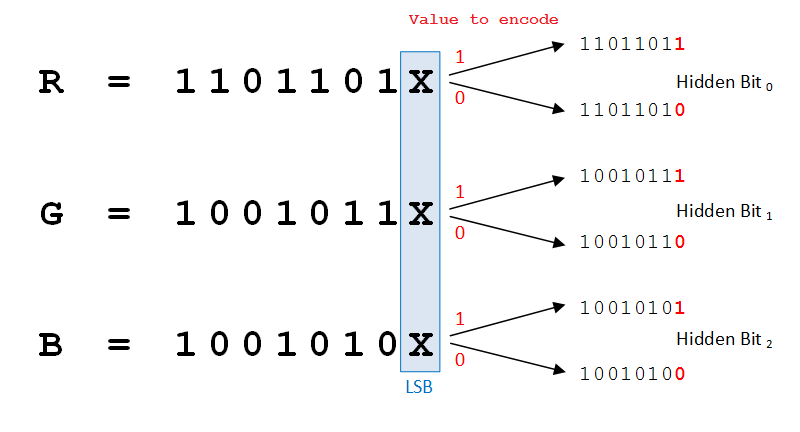


Fig: LSB Technique for Steganography

**5. RESULTS:**

5.1)Snapshots:

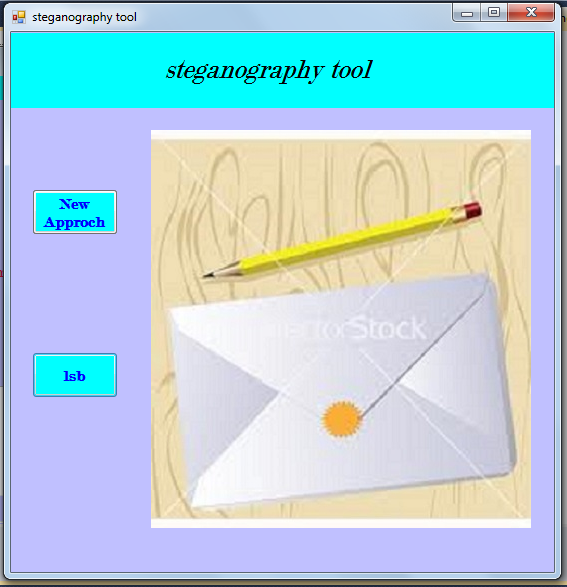


Fig 5.1.a: Main Form Display

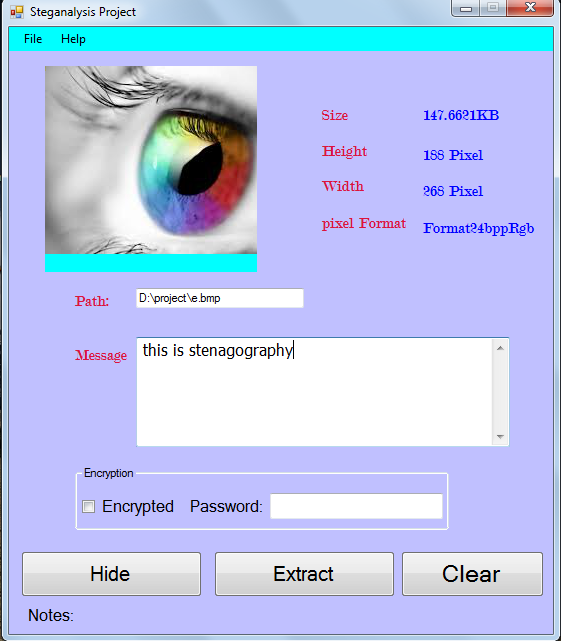


Fig 5.1.b:Loading an Image And writing secret Message

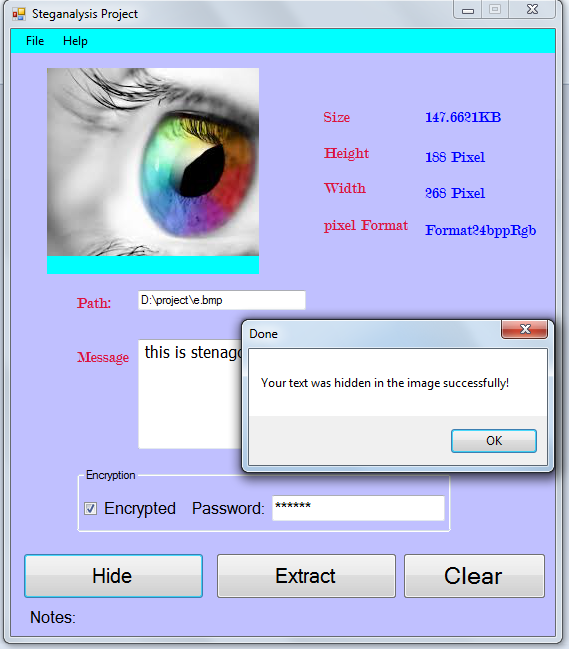


Fig 5.1.c: Encrypt Text with password protection

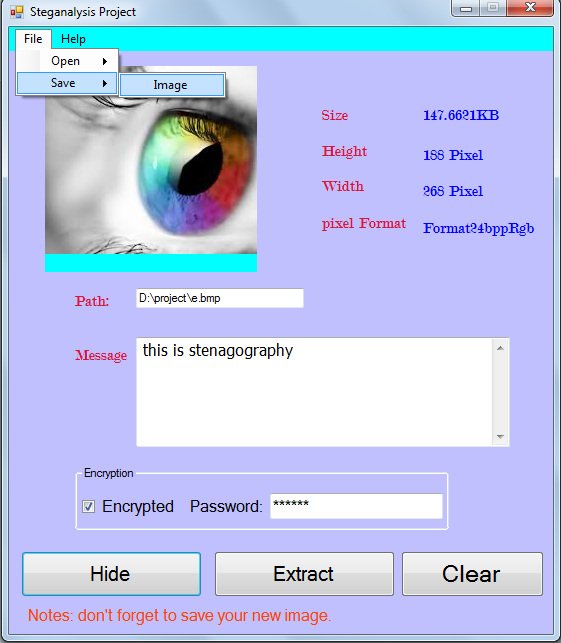


Fig 5.1.d: Saving the Stego-Image To location of our choice

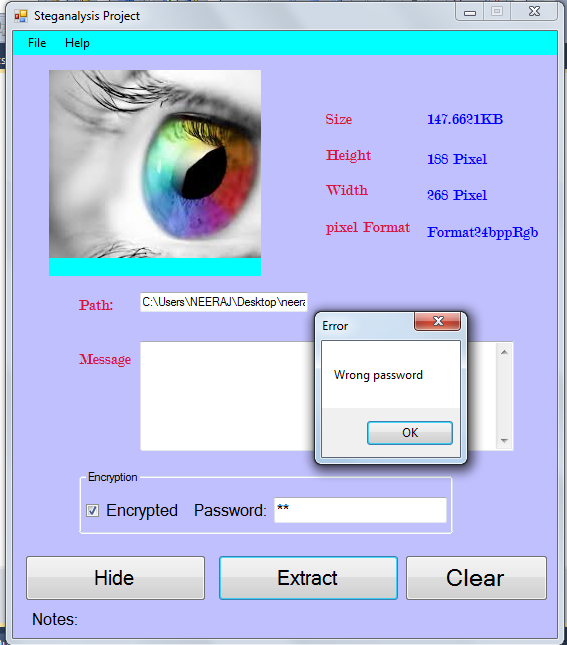


Fig 5.1.e: No Message decrypted when wrong password is Typed

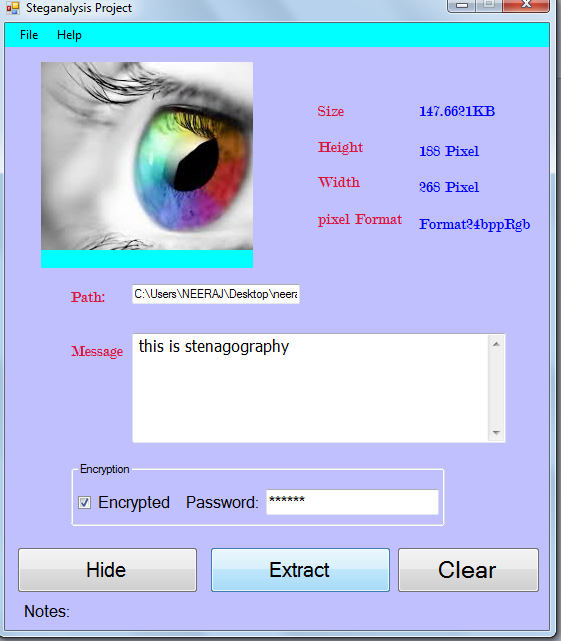


Fig 5.1.f: Decryption part working – Retrieves the hidden information

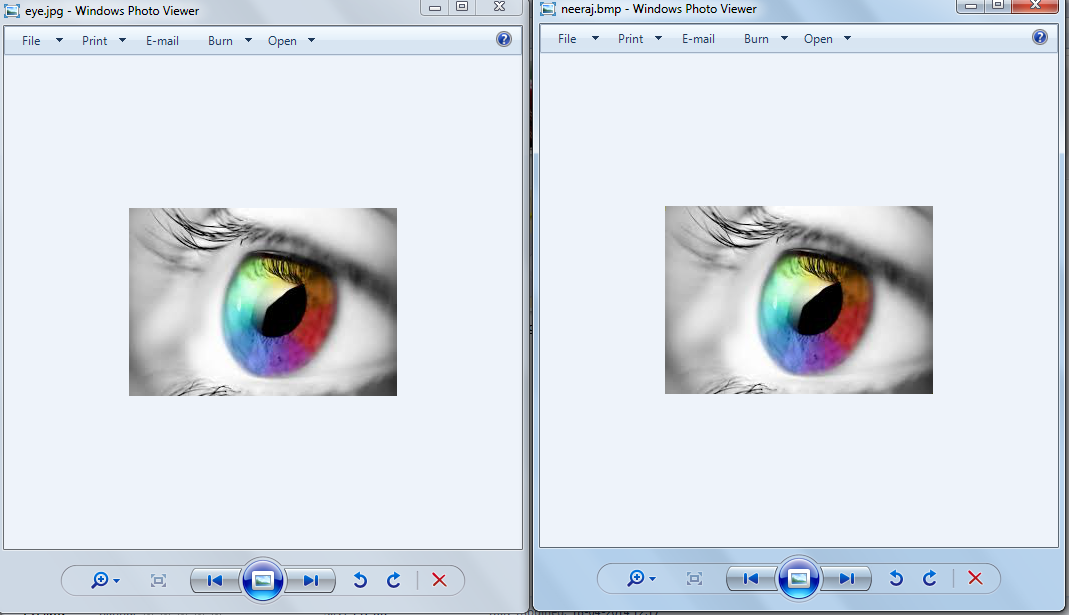


Fig 5.1.g: Original Image (eye.jpg) and stego-image (neeraj.bmp) Kept side by side

**5.2. Testing Inputs :**

|  |  |
| --- | --- |
| Sl No. of test case : | 1 |
| Name of test : | Steganalysis Test |
| Image Type being tested : | JPEG Image |
| Sample Input : | Sample Image, Text=”steganography” password=”naveen” |
| Expected output : | Encrypted Text=”steganography” password=”naveen” |
| Actual output : | Encrypted Text=”steganography” password=”naveen” |
| Remarks : | Success |

|  |  |
| --- | --- |
| Sl No. of test case : | 2 |
| Name of test : | Steganalysis Test |
| Image Type being tested : | PNG Image |
| Sample Input : | Sample Image, Text=”Hello World” |
| Expected output : | Encrypted Text=”Hello World” |
| Actual output : | Encrypted Text=”Hello World” |
| Remarks : | Success |

|  |  |
| --- | --- |
| Sl No. of test case : | 3 |
| Name of test : | Steganalysis Test |
| Image Type being tested : | BMP Image |
| Sample Input : | Sample Image, Text File=naveen.txt password=”neeraj” |
| Expected output : | Encrypted Text=naveen.txt password=”neeraj” |
| Actual output : | Encrypted Text=naveen.txt password=”neeraj” |
| Remarks : | Success |

**6.CONCLUSION:**

The main purpose of the project Steganalysis Testing Tool is to Hide information (usually Text) within an image and make confidential Transactions smoother. Using Steganalysis we can detect Hidden texts within images.

The Steganalysis Testing Tool is aimed to have the following features:

* Act as basic Encryptor
* Allow Password Encryption Facility to stego-image
* Display all the necessary Image information
* Act as a basic Decryptor

We started by providing an overview on the Concepts of steganography, before reviewing a wide range of steganographic techniques in detail . For the two algorithms that

was mentioned, an emphasis was put on the implementation such that we presented the

techniques from the viewpoint of steganographers.

**6.1 Limitations**

Though the project is a well-modulated system, it has been limited to certain restrictions:

* Cannot encrypt and decrypt for all Image types
* The size of the data to be stored within an image is restricted(few kb).
* Encryption and Decryption Works correctly only on characters that have ASCII values.
* Cannot store any other information apart from Text,pdf files.

**6.2 Future Enhancements**

Some enhancements which can be done on this project over time are:

* Can check the working of application for all types of images
* Design a stegalyser toolkit (like the ones used by Sarc in DOD)
* We can implement Encryption and Decryption of small audio,video and other files within images.
* Implementation of other efficient techniques.

**7. References:**

1) S. Dumitrescu, X. Wu, and Z. Wang. "Detection of LSB Steganography via Sample

Pair Analysis", Lecture Notes in Computer Science, vol. 2578, pp. 355-372, 2003.

2) (IJCSIS) International Journal of Computer Science and Information Security,

Vol. 7, No. 2, 2010 Khan Farhan Rafat

Department of Computer Science

International Islamic University

Islamabad, Pakistan

3) “Image steganography and steganalysis”

Author:Philip Bateman Department of Computing Faculty of Engineering and Physical Sciences

cs31pb@surrey.ac.uk

4) “Investigation and Experimentation in Digital steganography

By Ioannis L .Kolaxis,Birmingham June 2002 i.kolaxis@ieee.org

5) “International Journal of advanced Research in computer science and software Engineering “ Research paper available online at www.ijarcsse.com