**BANSILAL RAMANATH AGARWAL CHARITABLE TRUST’S**

**VISHWAKARMA INSTITUTE OF TECHNOLOGY**

**PUNE – 411037**

****

**PROJECT REPORT**

**SteganoCrypt Tool**

**GUIDE**

**PROF. M.Gandhi**

**PRESENTED BY**

**TE-COMP-I**

**ROLL NOS: 19,22,28,60,61**

**2011-2012**

**BANSILAL RAMANATH AGARWAL CHARITABLE TRUST’S**

**VISHWAKARMA INSTITUTE OF TECHNOLOGY**

**PUNE – 411037**

****

**CERTIFICATE**

This is to certify that the **Project titled**

**SteganoCrypt Tool**

Has been completed in the academic year 2010 – 2011 by

|  |  |
| --- | --- |
|  |  |
|  |  |

in partial fulfilment of the bachelor’s Degree in **COMPUTER** Engineering .

Date: **GUIDE** Head of the department

**Abstract**

Three different aspects in information-hiding systems contend with each other: capacity, security, and robustness. Capacity refers to the amount of information that can be hidden in the cover medium, security to an eavesdropper’sinability to detect hidden information, and robustness to the amount of modification the stegomedium can withstand before an adversary can destroy hidden information.

Information hiding generally relates to both watermarking and steganography.

Steganography, or “hidden writing” can be traced back to 440 BC in ancient Greece. Often they would write a message on a wooden panel, cover it in wax, and then write a message on the wax. These wax tablets were already used as writing utensils, so the hiding of a message in a commonly used device draws very little suspicion. In addition to use by the Greeks, the practice of steganography was utilized by spies in World War II.There were even rumors that terrorists made use of steganography early in 2001 to plan the attacks of September 11

The goal of steganography is to hide messages inside other harmless messages in a way that does not allow any enemy to even detect that there is a second secret message present. The only missing information for the enemy is the short easily exchangeable random number sequence, the secret key, without the secret key, the enemy should not have the slightest chance of even becoming suspicious that on an observed communication channel, hidden communication might take place.

In an ideal world we would all be able to sent openly encrypted mail or files to each other with no fear of reprisals. However there are often cases when this is possible, either because the working company does not allow encrypted email or the local government does not approve of encrypt communication (a reality in some parts of the world). This is where steganography can come into play.

**Introduction**

Steganography comes from the Greek and literally means, "Covered writing". It is one of various data hiding techniques, which aims at transmitting a message on a channel where some other kind of information is already being transmitted. This distinguishes steganography from covert channel techniques, which instead of trying to transmit data between two entities that were unconnected before.  
Steganography has developed a lot in recent years, because digital techniques allow new ways of hiding informations inside other informations, and this can be valuable in a lot of situations. The first to employ hidden communications techniques -with radio transmissions- were the armies, because of the strategic importance of secure communication and the need to conceal the source as much as possible.  
Nowadays, new constraints in using strong encryption for messages are added by international laws, so if two peers want to use it, they can resort in hiding the communication into casual looking data. This problem has become more and more important just in these days, after the international Wassenaar agreement, with which around thirty of the major - with respect to technology - countries in the world decided to apply restrictions in cryptography export similar to the US ones.  
Another application of steganography is the protection of sensitive data. A file system can be hidden in random looking files in a hard disk, needing a key to extract the original files. This can protect from physical attacks to people in order to get their passwords, because maybe the attacker cant even know that some files are in that disk.   
The major concern of steganography is stealth, because if an attacker, either passive or active, can detect the presence of the message, from that point he can try to extract it and, if encrypted, to decrypt it. The resistance to attempt at destruction or noise is not required, since we consider the sender and the receiver equally interested in exchanging messages, so that they will try to transmit the stego-medium in the best way they can. If the stego-data can be transmitted over the selected channel, and this is usually the case with all the media that are used, like images or sounds, then the embedded data will be preserved along with them. Thus, data hiding techniques for steganography must focus on the maximum strength against detection and extraction.  
As a second request, we would prefer a high data rate, because we will usually want to be able to exchange any amount of data, from simple messages to top secret images.

People use cryptography to send secret messages to one another without a third party overseeing the message. Steganography is a type of cryptography in which the secret message is hidden in a digital picture. Think of all those pixels in an image and each pixel has three color numbers — there are zillions of numbers in an image. If you were to change a few of these color numbers the resulting picture would probably look a lot like the original image; in fact, most people probably couldn’t tell that you had changed the image at all. Steganography works by changing a few pixel color values; we will use selected pixel values to represent characters instead of a color value. Of course, the resulting image will still look mostly like the original except that a few tiny ”blips” might seem a little out of place if you look very closely.

We can then send the image to a buddy and they can extract the message if they know which pixels to decode.

In this project we will be writing a java application that will enable you to exchange secret messages with another person.

***Bitmap Files***

• First we will need to read your picture as a jpg and then save it in 24-bit bmp format. You will need to use bmp files for this assignment because jpg’s are ”lossy” meaning that what you write to the file may be changed slightly so that the resulting image can be stored more efficiently. Thus jpg will not work for steganography because jpgs will change the secret message when storing the file to disk.

**Design Details**

Here are some of the skills you will need in order to complete this assignment:

• Loading and saving bitmap files.

• Bit manipulation operations.

We have to manipulate the bits stored in numbers. There are three basic bit manipulation operations: and, or, and shift. We will be using all of the three.

The most basic design implementation problem is –

How are we going to hide the data file behind the images i.e bitmap images ?

It means how are we going to hide the characters in the text file behind the pixels in the bitmap image?

The Solution for it is –

In a 24 bit bitmap image format each pixel is having 24 bit length where these 24 bits are divided into 3 bytes containing 8 bit each.The first byte corresponds to red color value , the second byte to green color value and the third to blue color value .The length all these values is 8 bit long ie these values can be between 0 to 255 . And combination of all these values generate desirable color for a given pixel

**Hiding methods**.

First we will extract character from the file ,then find its ascii value and then convert that ascii value .Our ascii value will be between 50 to 150(generally) and pixel having 3 values ranging from 0-255 for red,blue green respectively.For hiding the ascii value we can use either of the three values i.e replace either of these three values by our ascii value.

The problem with changing the red values in our encode/decode steps, is that these often cause quite visible changes in the resulting image. This is especially true if the pixels that are being changed are part of a large section of uniformly colored pixels – the ”dots” stand out and are noticeable. As an option, we can change only the lower order bits of each pixel color (red, blue, and green). This will make subtle changes to each pixel’s color and will not be as evident.

Remember that each pixel has three bytes: one byte for red, blue and green colors. Each byte has 8 bits to encode a number between 0 and 255. When we swap out the red color byte for a character, it is possible that we are changing the redness of that pixel by quite a bit. For example, we might have had a pixel with values of (225, 100, 100) which has lots of red, some green and some blue – this is basically a reddish pixel with a slight bit of pink color to it. Now suppose we are to store the character ”a” in the red part of this pixel. An ”a” is encoded as decimal number 97 so our new pixel becomes (97, 100, 100). Now we have equal parts ofall three colors to produce a dark grey pixel. This dark grey is noticeably different

than the dark pink we had before; it will definitely stand out in the image especially if the other nearby pixels are all dark pink. We want a way to encode our message without making such drastic changes to the colors in the original image. If we only change the lowest bits of each pixel, then the numeric values can only change by a small percentage. For example,

suppose we only change the last three bits (lowest three bits) – these are the bits that determine the ”ones place”, the ”twos place” and the ”fours place”. We can only alter the original pixel color value by ±7. Let us think of our original pixel as a bits:

(r7 r6 r5 r4 r3 r2 r1 r0, g7 g6 g5 g4 g3 g2 g1 g0, b7 b6 b5 b4 b3 b2 b1 b0).

And our character (byte) as some bits:

c7 c6 c5 c4 c3 c2 c1 c0.

Then we can place three of these character bits in the lowest red pixel, three more in the lowest green pixel, and the last two in the lowest blue pixel as follows:

(r7 r6 r5 r4 r3 c7 c6 c5, g7 g6 g5 g4 g3 c4 c3 c2, b7 b6 b5 b4 b3 b2 c1 c0).

If we had done this to the example of pixel (225, 100, 100) with character ”a”,

we obtain:

original pixel = ( 11100001, 01100100, 01100100 )

”a” = 01100001

new pixel = ( 11100011, 01100000, 01100101 )

new pixel = ( 227, 96, 101 )

Notice the new pixel of (227, 96, 101) is almost the same value as the old pixel of (225, 100, 100). There will be no noticeable color difference in the image! To retrieve the message, you simply extract the appropriate pixels from the rgb values to reconstruct the secret character.

**Implementation details**

***As per our GOAL we have implemented a strong code so that if the attackers want to know the message even if he has the target image containing a data .He will not be able to decrypt it in polynomial time.***

For the implementation of this stegano project we have used java platform jdk1.6.

Firstly we have created a graphical user interface wherein the user will be able to interact with the system desingned.In this graphical interface we have get some choices for the user to click.(as per his interest)

The three main choices are

1.Upload – The user need to upload a Image (Bitmap image strictly) on which he wants to apply the concept of steganography (hide a data)

2.Hide- Using the hide interface the user will get a dialog box with a text field in it .In that text field the user has to enter message which he or she wants to hide.

3.Extract – After pressing extract button user will get a open dialog box,where he need to select the file in which he has hidden the data.

As soon as he selects the file he will get the message from the file.

Classes used

Images class- Here we have built the actual graphical interface for our entire project

HideDlg class – Here we create a interface for writing a message

GetPixelColor- Here we implemented the actual logic behind the entire processing involved in hiding of the data behind the selected file.

Here we also provided the logic for extracting of data from file.

**Future Works**

We have to achieve the following goals in our future implementation

1.Security of the software

2.Apply following cryptographic algorithms for the data

DES

RSA

SHA

3.And also encrypt the sequence of choosing the pixels order

**Result**

1.The program is tested for various character strings which are hidden into a image

2.The two images ie the original and the image which contain actual message cannot be differentiated with our eyes.