



Image steganography using C#

Shivam Kaul¹, Sanchit Gupta², Syed Abbas Haider³

^{1,2,3}Student, IMS Engineering College, Ghaziabad, Uttar Pradesh

ABSTRACT

This aim of this paper is to give an overview of Image Steganography in which the hidden information can be embedded in the image, so the security of the information is not violated. We have made the application using Visual studio and coding part is done using C#. LSB algorithm is used to make sure the secure communication is taking place. The main purpose of this is to maintain secure communication between two parties so that no information is lost.

Keywords— Stego-Image, Steganography, Cover-object, Stego-key, LSB

1. INTRODUCTION

First of all, we know the importance of steganography. Image Steganography refers to the process of hiding data within an image file. The image selected for this purpose is called the cover-image and the image obtained after steganography is called the stego-image.

Here in this, the application is made with the help of Visual Studio and the coding part is done in C# language.

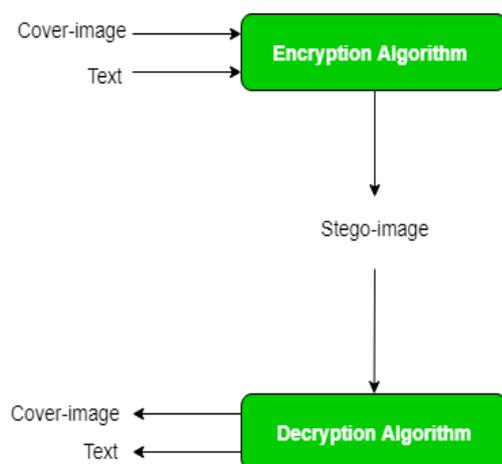


Fig. 1: Image steganography

2. PROCEDURE

An image is represented as an $N \times M$ or $N \times M \times 3$ matrix in memory, with each entry representing the intensity value of a pixel. In image steganography, a message is embedded into an image by altering the values of some pixels, which are chosen by an encryption algorithm. The recipient of the image must be aware of the same algorithm in order to know which pixels are actually needed to start the process.

Images are made up of pixels which usually refer to the colour of that particular pixel. In a greyscale (black and white) image, these pixel values range from 0-255, 0 being black and 255 being white.

The image would be encrypted and that can be of any image file type because the application itself would convert the image to BMP Format by itself.

2.1 Concept of LSB

LSB stands for Least Significant Bit. The idea behind LSB embedding is that if we change the last bit value of a pixel, there won't be much visible change in the color. For example, 0 is black. Changing the value to 1 won't make much of a difference since it is still black there would be a slight difference, but that does not matter.

- Convert the image to Greyscale.
- Resize the image if needed.
- Convert the messages to its binary format.
- Initialize the output image the same as that of the input image.
- Traverse through each pixel of the image and do as follows from below:
 - Convert the pixel value to binary.
 - Get the next bit of message to be embedded.
 - Create a variable temp.
 - If the message bit and LSB of the pixel are same, set temp = 0
 - If the message bit and LSB of the pixel are different, set temp = 1
 - The setting of temp can be taken by taking XOR of message bit and LSB of the pixel.
 - Update the pixel of the output image to input image pixel value + temp.
- Keep updating the output image till all the bits in the message are embedded.
- Finally, write the input as output as well.

2.2 Concept of C#

C# is a simple, modern, general-purpose, object-oriented programming language developed by Microsoft within its .NET initiative led by Anders Hejlsberg. C# is a modern, general-purpose, object-oriented programming language developed by Microsoft and approved by the European Computer Manufacturers Association (ECMA) and International Standards Organization (ISO). C# is based on object-oriented programming concepts.

2.3 Concept of visual studio

Visual Studio .NET is a Microsoft-integrated Development Environment (IDE) that can be used for developing consoles, Graphical User Interfaces (GUIs), Windows Forms, Web services and Web applications. Built-in language services offer support for different well-known languages (like C#, C++, VB, and F#), while language services may be installed separately to support languages, including M, Python, and Ruby, among other supported languages.

3. METHODOLOGY

User needs to run the application. The user has two tab options: encrypt and decrypt. If user select encrypt, application gives the screen to select image file, information file and option to save the image file. If user select decrypt, the application gives the screen to select the only image file and ask path where user want to save the secrete file.

- This project has two methods: Encrypt and Decrypt.
- In encryption, the secret information is hiding in with any type of image file.
- Decryption is getting the secret information from the image file.

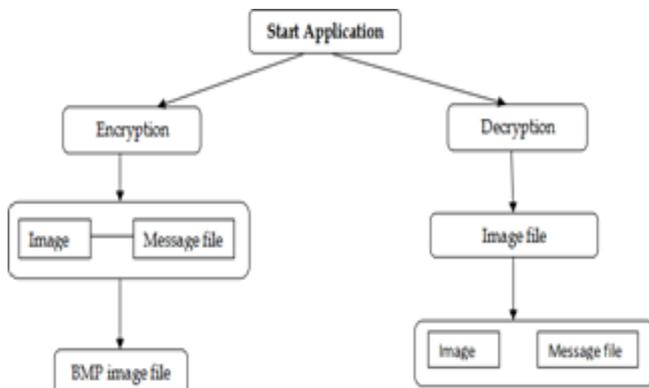


Fig. 2: Application model

4. RESULT AND ANALYSIS

So now we see the two images which look like each other but in actual they are different because one of them is containing the hidden information within it. First one is the original image which is been chosen for the process.

With the image, we have shown the properties of the image also, so as there won't be any confusion regarding the image itself. During encryption, the image is loaded of any format, now comes the option for loading the information file within the image itself. Now user can load whatever file it has to be encrypted after it is loaded the encryption would be done, and ht image file would be saved in the form of BMP format.

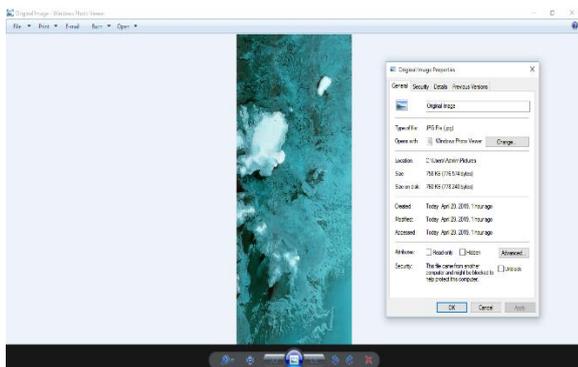


Fig. 3: Original image

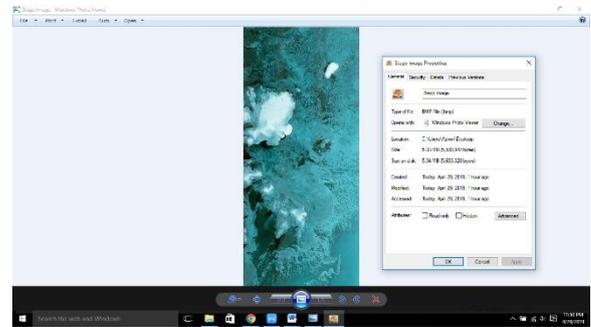


Fig. 4: Stego - Image

Now when the decryption would start, it would load that particular Stego- Image and would decrypt it. During this process, the application would ask the user to save the information file to some location, after that the information file would be saved in the desired location and stego image would remain the same.

5. CONCLUSION

Steganography can be used for hidden communication. We tried to make sure the whole working of the application would be done in a way that it would be concise, fast, error-free and user-friendly.

LSB Algorithm proved the secrecy of the Information file within the image itself, which would make sure there won't be any kind of information loss and it is useful in a way that any type of file can be encrypted within the image.

Here are some snaps of Encryption and Decryption.

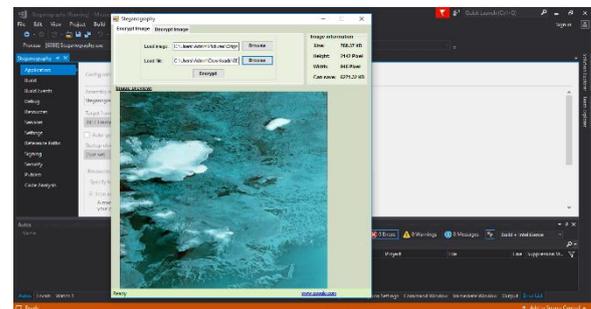


Fig. 5: Encryption

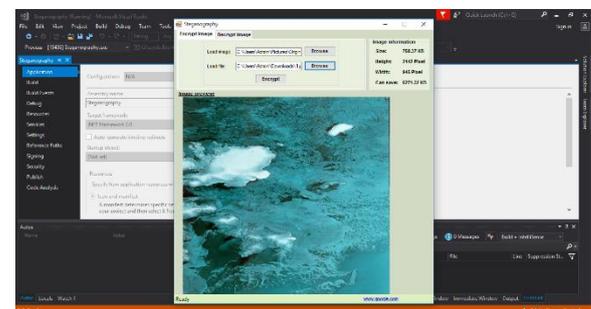


Fig. 6: Decryption

6. REFERENCES

- [1] <http://www.google.com>
- [2] <http://www.codeproject.com>
- [3] <http://www.asp123.com>
- [4] <http://www.wikipedia.org>
- [5] Mastering C# (Paperback).
- [6] A Catalogue of Steganography tool.
- [7] <http://www.GeeksforGeeks.com/reference>.