# Interim Specification

Introduction

Steganography is the concept of hiding data in plain sight. Digital steganography is the successor to physical methods used in the past such as the use of secret inks to write messages on paper with otherwise innocuous messages written over them or writing messages on the scalp of a messenger which are then hidden by the hair.

In this project, a program will be created that will hide information within the user’s chosen image file. The program will then also be able to retrieve this information and save it separately in the desired file format of the user. If time permits, encryption of the data to be hidden will be investigated and the creation of a program to search for hidden data will also be considered.

The most appropriate file type for this application is the .bmp file type. The .jpeg file type is unsuitable for this application due to its aggressive compression of files. The use of .jpeg would likely cause the data hidden within the file to become corrupted or unrecoverable. There are some other file types that could be used for this application such as .tiff and .png files, however, in addition to the RGB values required for the image analysis, these file types also contain an alpha value. This alpha value is responsible for dictating the opacity of the pixel, this extra information would increase the complexity of the information to be hidden, making the embedded data more vulnerable to discovery.

Various stenographic techniques are used for hiding files within images, these techniques include; least significant bit steganography is the simplest and most common stenographic technique used for image steganography. The technique works by taking a cover image and hiding another Image within the least significant bits of the pixels of the cover image, this in turn creates a new stenographic image which contains both images however only one is visible to an observer. However, this method is susceptible to a “visual attack” where upon a close examination an observer can notice the noise in the image, making it the most exploitable method.

Another method, which can be used, for image steganography is discrete cosine transform steganography, this method works by taking a JPEG image and separating the image into parts of differing importance, this separates the image into high, middle and low frequency. The hidden image is then hidden within these frequencies; this allows the image to be hidden and is more effective than LSB steganography as it is not exploited using a “visual attack”

Program Function

Figure 1 below shows the proposed class hierarchy for the program.

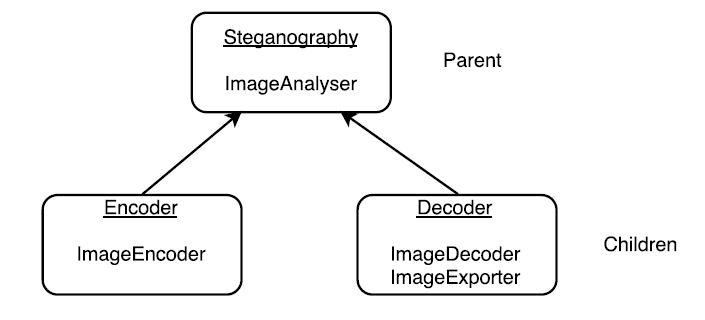


Figure 1 Program class hierarchy

A brief outline of each function detailed within Figure 1 can be found below:

* ImageAnalyser – Converts a 256 colour bitmap into a binary sequence to be stored in a ‘pickle’ object or a multidimensional array (to be decided).
* ImageEncoder – Hides the binary sequence of the data to be hidden within the two least significant bits of the cover image.
* ImageDecoder – Converts the binary sequence of the stego object into two separate binary sequences (the cover and the hidden data).
* ImageExporter – Converts a binary sequence into a 256 colour bitmap.

Both child classes inherit the ImageAnalyser function from the parent class.

To hide the data within the cover Image the following will be followed for the code:

* Create a bespoke function called ImageAnalyser which calls a carrier image and an outfile
* Open image and convert to RGB
* Get dimensions for image (width and height)
* Create array and store pixel RGB information within it (empty initially)
* Create a loop to get RGB value each pixel for x and y within image dimensions
* Add each RGB value to array by using append function
* Convert RGB value of each pixel to a binary pickle stream
* Add the data from the new array to the image
* Create an instance of the function Image analyser

\*\*Describe method of separating the cover image and the hidden data i.e. changing the RGB values of the stego object’s pixels by 3? \*\*

Figure 2 below shows the proposed flowchart for the encoder class.



Figure 2 Encoder class flowchart

Figure 3 below shows the flowchart for the decoder class.



Figure 3 Decoder class flowchart

Project Timeline

A Gantt chart was created and can be found below in Table 1 detailing the stages in the project and their target completion dates.

Table 1 Steganography Program Gantt chart



The tasks detailed above were split up as follows:

Josh Ward: Steganography Techniques, Data Hiding Program, Encryption Algorithm.

James Wilcox: File Type Suitability, Data Retrieval Program, Steganalysis Program.

Combined: Testing Phase, Spec (Interim Specification), Report + Demo.

Software Testing Procedure

\*\*Describe the proposed method of testing i.e. stress testing (large file size test etc.), tested against existing products. \*\*