1. **Overview**

   The program will allow a user to insert a hidden message inside a windows bitmap image file. The techniques used are known as image steganography.

2. **Problems Addressed**

   The program solves the problem of a user wishing to transmit or store a message in a way such that no one but the user will know of that message's existence.

   Steganography is fundamentally different from a more well-known aspect of computer security: the art of cryptography. With cryptography it's often the case that an adversarial party knows that a message exists and may be able to intercept it. He or she may never actually read it, but that cannot be assured when the techniques of cryptanalysis are applied by a determined foe.

   However, with steganography, the very fact of a message's existence remains unknown to the foe. It is hidden in ordinary network traffic or files local to a computer system, and ideally, has no effect on a users perception of that file. In other words, the images that will be created by this program will *look* exactly the same as an untreated image and if intercepted will give no clue that a hidden message is present.

   This supplies an additional level of security for private information or messages. When combined with encryption and text deflation (that is, *"zipping"* a text file), a very high degree of security can be attained.

3. **Interface**

   The program presents a small window to the user presenting three buttons indicating the three main options of the program:

   (a) create a steganographic image,

   (b) recover a hidden message from a steganographic image, or

   (c) exit.

   Button (a) will take the user to a window where the following must be entered:

   (I) An image on the local file system to apply the message to.

   (II) A text file on the local file system that will be applied to the image.

   (III) *(optional)* Whether to encrypt the text file using a user supplied pass-phrase.

   Additionally, buttons will be present to allow the user to cancel the operation (returning to the initial window), or to apply the options, once the required information is entered.

   (I) is a button that opens a file-selection dialog where the user chooses an image. (II) is a button that opens a file-selection dialog where the user chooses a text file containing the message. (III) is a check-box which when checked will present a small dialog box

where the user must type in a plain-text pass phrase. The *'Apply'* button will create the steganographic image and immediately open a file-selection dialog to save the new image to the local file system.

Button (b) will take the user to a window where the following must be entered:

(A) An image on the local file system to extract the message from.

(B) The name of a text file to save the message to.

(C) *(optional)* Whether the message is encrypted.

Additionally, buttons will be present to allow the user to cancel the operation (returning to the initial window), or to apply the options, once the required information is entered.

(A) is a button that opens a file-selection dialog where the user chooses an image file that has a message embedded. (B) is a button that opens a file-save dialog where the user enters a filename and path to save the message to. (C) is a check-box which when checked will present a small dialog box where the user must type in a plain-text pass phrase. The program will extract an encrypted message regardless of whether this box is checked, however. The *'Apply'* button will extract the message and save it to the text file.

Button (c) will cause the program to halt.

4. **Further Details**

   (a) **Inputs** This program will only process ASCII text as inputs. It is not designed to process multi-byte text formats such as Unicode for messages or pass-phrases, though it is possible that multi-byte pass-phrases *may* work. Additionally, the program will only process images files in the device-independent bitmap format as defined by the Microsoft company. Such files are usually identifiable by their *.bmp* file extension. The program will not be able to process compressed image formats such as JPEG, PNG, etc.

   (b) **Outputs** This program will output files in the formats listed in (a).

   (c) **Error Handling** The program will not go to great lengths to handle errors. If a user attempts to open an image file that is of the wrong type, the program will likely fail. If a user attempts to extract a message from an image that has none, the saved message will probably be blank.