**Project 3 - Steganography**

**Programming Language: Python 3**

**Executive Summary:**

The project aims at embedding a secret message into the given image without creating any visible changes like change in color or size of the carrier image and also to extract the embedded message from the image again.

The deliverable is a python script called ‘Stegun’, which can secretly embed the given message into the given image and can also extract the message out from the given image.

**Technical details:**

The program Stegun.py provides the user with two options, one for embedding the message and the other for extracting the message.

=> When the first option is selected, the program prompts the user to enter the secret message. The program only supports messages with length lesser than 95 bytes, since when the message exceeds that size, it will be very harder to embed it without creating any visible changes.

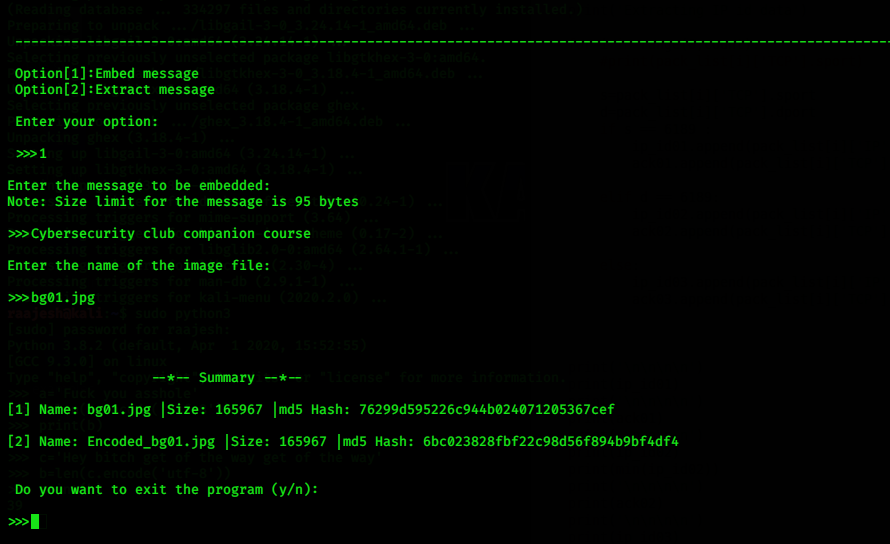
=> Then the program asks the user for the carrier image’s name with the proper path.

=> After that the size and md5 hash of the image is calculated. Function hasher is used for hashing.

=> Then the message is embedded into the carrier message and the entire data is copied to a new file, that will be named based on the original name. ( Encoded\_<original name>.jpg)

=> And finally, the name, size and md5 hashes of both the original image and the encoded image are displayed as summary.

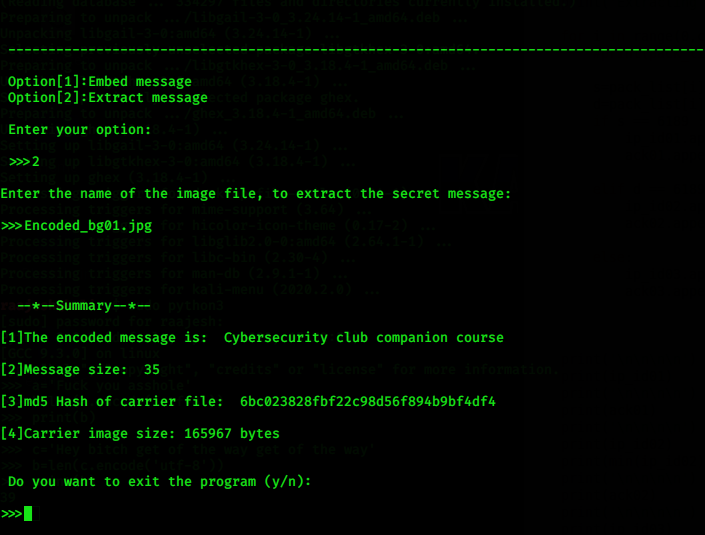




=> When the second option is selected, the program asks for an encoded image file, ( however the program only accepts carrier files created from the program itself ).

=> Then, the message is obtained from the image file, before which the size and md5 hash of the image file is calculated.

=> And, finally the message and the data about the carrier file are provided as summary.



=> From the given images, it can be found that the size of both the original and the encoded images are the same.