Ashish Adhikari
@02746154

Computer Security 2
Steganography for BMP image files implemented in C

- **Steganography:**

  Steganography is used to hide secret messages into media files that are undetectable by normal human observation. This project is one such approach to encode secret messages into BMP image files to make it indistinguishable to the human eye without altering the file size.

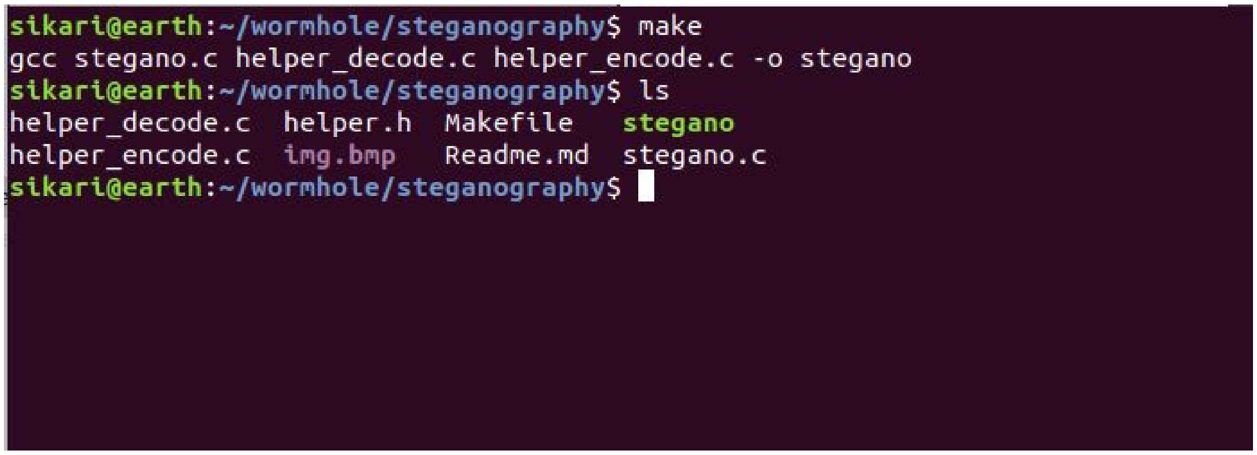
- **Instructions to build and run: (also available in Readme.md file and screenshot of Readme.md)**
  - **To compile:**
    - run make
  - **To encode into image** *img.bmp* **while saving message to** *secret.txt:*
    - ./stegano -E -i img.bmp -m secret.txt
    - Note that this saves the modified image as cipher-img.bmp for img.bmp file. I.e. it attaches “cipher-” prefix to the original image and saves.
  - **To decode message from a cipher image:**
    - ./stegano -D -i cipher-img.bmp -m secret.txt

- How it works:
  - Encryption:
    1. First 8 bytes (after the 54 bytes mandatory BMP) are used to encode the message length using function size_encrypt(). This is described as follows:
    2. If least significant bit (LSB) for the byte and the running variable shift (decreasing shift rights) match, the byte is written into cipher image as it is
    3. Else if the (LSB) is 0, the (LSB) is flipped to 1 in the new cipher image
    4. Else the result of XOR of current byte with 1 is placed as the (LSB) in the new image file.
    5. Message is also encrypted in the same manner as 1-4 for all bytes in group of four bytes until the end of file is reached in the secret.txt file. After this, if EOF is note reached, remaining bytes are copied as is into the new image file
  - Decryption:
    1. If (LSB) is 1, shift left the byte by 1 and set (LSB) as 1. Else, shift left the byte by 1. This will give us the size.

2. Repeat 1 for 8 byte group and write to image file in every iteration.

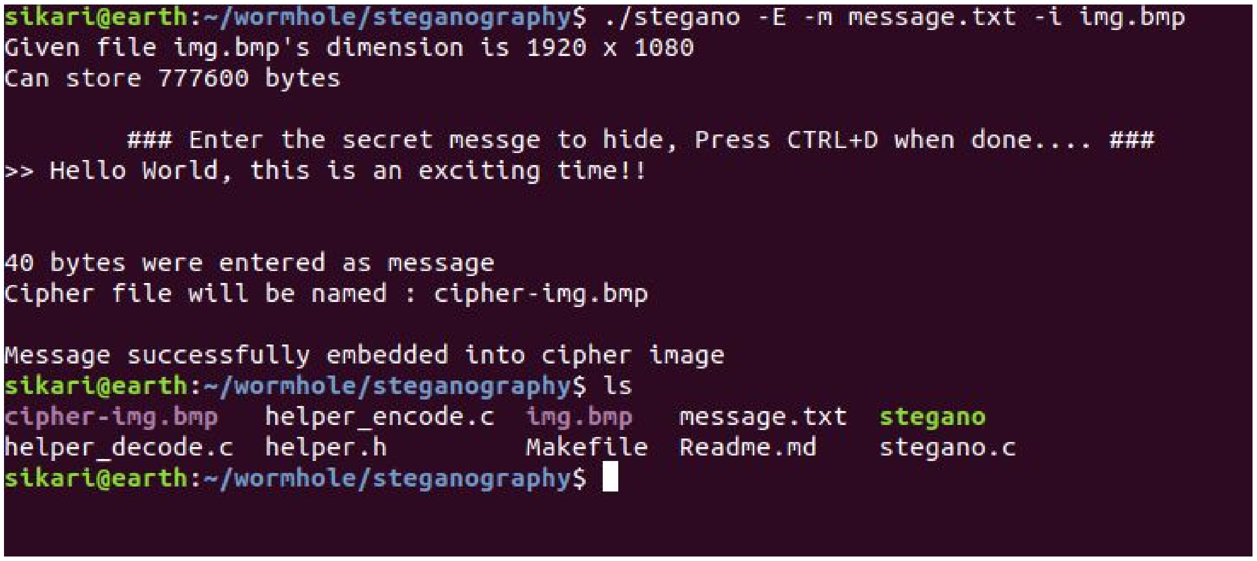
Sample run:

1. Build using make



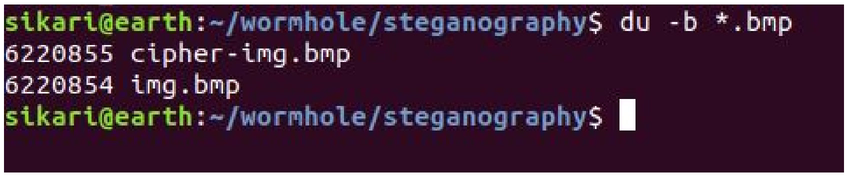
```
sikari@earth:~/wormhole/steganography$ make
gcc stegano.c helper_decode.c helper_encode.c -o stegano
sikari@earth:~/wormhole/steganography$ ls
helper_decode.c  helper.h  Makefile   stegano
helper_encode.c  img.bmp   Readme.md  stegano.c
sikari@earth:~/wormhole/steganography$ 
```

2. Encode message “Hello World, this is an exciting time!!”

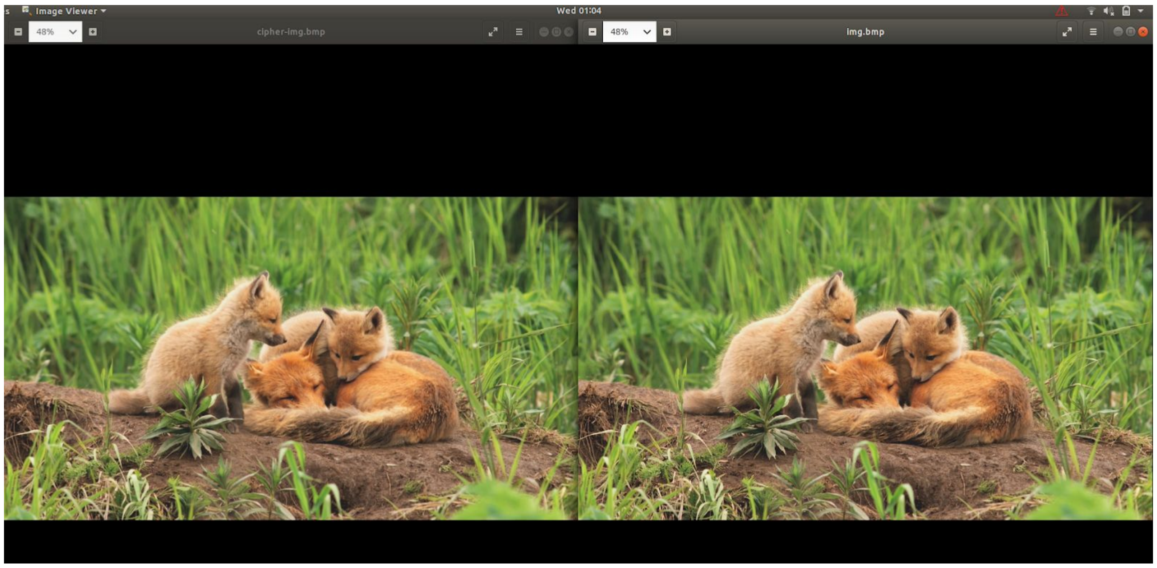


```
sikari@earth:~/wormhole/steganography$ ./stegano -E -m message.txt -i img.bmp
Given file img.bmp's dimension is 1920 x 1080
Can store 777600 bytes

        ### Enter the secret messge to hide, Press CTRL+D when done.... ###
>> Hello World, this is an exciting time!!


40 bytes were entered as message
Cipher file will be named : cipher-img.bmp

Message successfully embedded into cipher image
sikari@earth:~/wormhole/steganography$ ls
cipher-img.bmp   helper_encode.c  img.bmp   message.txt  stegano
helper_decode.c  helper.h         Makefile  Readme.md    stegano.c
sikari@earth:~/wormhole/steganography$ 
```

3. Make sure that original and cipher image do not look different and are similar size

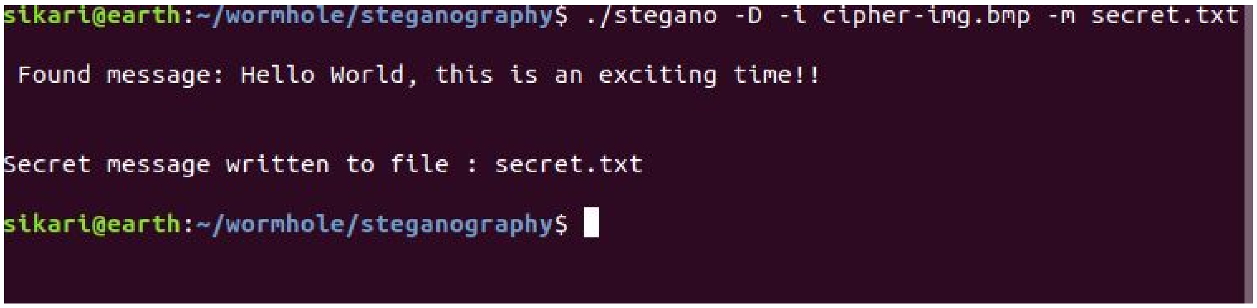


```
sikari@earth:~/wormhole/steganography$ du -b *.bmp
6220855 cipher-img.bmp
6220854 img.bmp
sikari@earth:~/wormhole/steganography$ 
```



Result - No visible and disk usage difference!!! (left cipher-img.bmp)

4. Decoding:



```
sikari@earth:~/wormhole/steganography$ ./stegano -D -i cipher-img.bmp -m secret.txt

 Found message: Hello World, this is an exciting time!!


Secret message written to file : secret.txt

sikari@earth:~/wormhole/steganography$ 
```

5. More documentation available in the Readme file

Steganography

Steganography is used to hide secret messages into media files that are indistinguishible to the normal observer

To build

```
$ cd steganography
$ make
```

Embedding message

```
$ ./stegano -E -i image.bmp -m message.txt
```

Decoding message

```
$ ./stegno -D -i cipher.bmp -m secret.txt
```

**Please note that order of the flag does not matter

Flags

```
-E = Encode message
-D = Decode message
-i = Original Image or Cipher Image
-m = message saved as text file
```

To remove generated binaries and media file

```
$ make clean
```