

IMAGE STEGANOGRAPHY USING PYTHON AND FLASK

Minor-II



Submitted By:

Isha Srivastava (9917102127)
Kushagra Gupta (9917102203)
Mukul Chauhan (9917102220)

Under The Supervision Of:

Dr. Kapil dev Tyagi

Department Of ECE
Jaypee Institute Of Information Technology
June, 2020

PROBLEM STATEMENT

In this digital era, the intensive collection of data and the inherent advantages of the new technology have spawned the cynical idea that privacy is dead. Moreover, recent instances of information leak by our beloved social networking sites put us at the risk of privacy breach. This image steganography web application is designed to relieve users of this data breach risk and instill a feeling of security amongst them.

OBJECTIVES

1. **SECURITY AND PRIVACY** : Unidentified transmission of critical data.
2. **ENCODING DATA INTO A PICTURE**: Parsing data into an image's pixels preserving the initial integrity of the image.
3. **DECODING DATA INTO A PICTURE**: Analysing an image for some hidden text message.
4. Reusability of our application: Web application based platform.

STEGANOGRAPHY

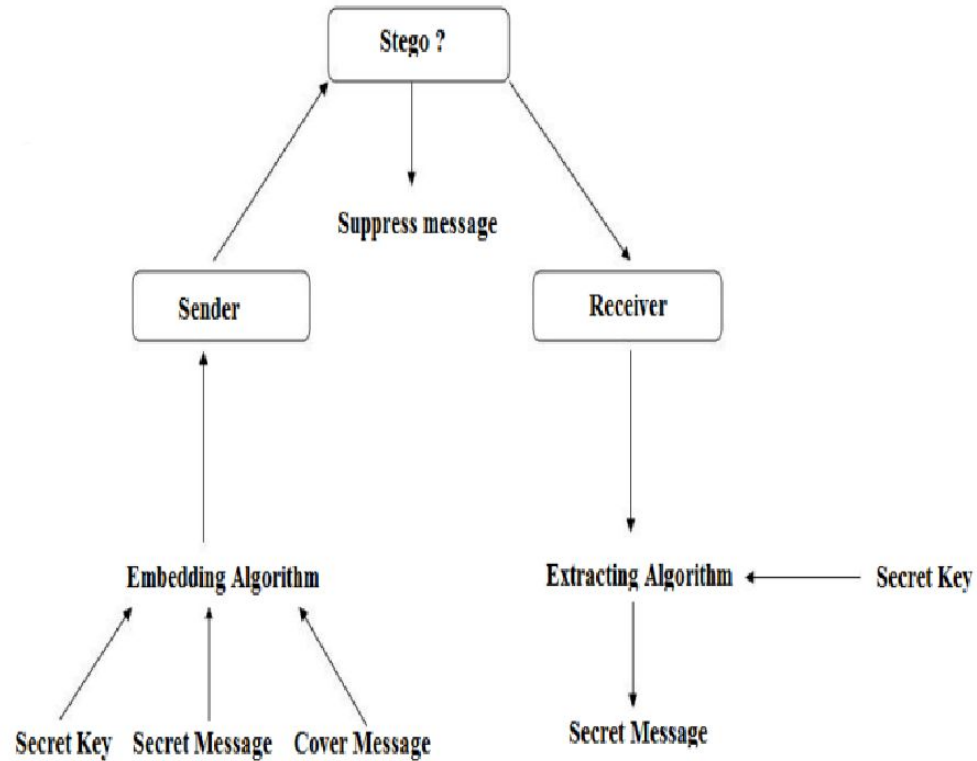
The above objectives can be achieved by using cryptography or steganography.

WHY STEGANOGRAPHY?

- Steganography is the science of writing hidden messages in such a way that no one except the sender and the receiver will be able to read
- The advantage of steganography over cryptography alone is that the intended secret message does not attract attention to itself as an object of scrutiny.
- For example, a sender might start with an innocuous image file and adjust the colour of every hundredth pixel to correspond to a letter in the alphabet. The change is so subtle that someone who is not specifically looking for it is unlikely to notice the change.

SOFTWARE USED

1. PYTHON
2. HTML
3. FLASK
4. CSS
5. BOOTSTRAP



GENERAL STEGANOGRAPHY APPROACH

METHODOLOGY

- **MATRIX REPRESENTATION:** An image is represented as an $N \times M$ or $N \times M \times 3$ matrix depending upon the colour in the memory, with each entry representing the intensity value of a pixel.
- **PYTHON SCRIPT USING EMBEDDING ALGORITHM:** Conversion of the images on the basis of modifications done to the LSB.
- **FLASK:** to give the script a web application functionalities, we have used flask that allows to render HTML pages and enable communication between script and HTML pages.

PYTHON SCRIPT USING EMBEDDING ALGORITHM

- Our algorithm involves the basic idea of changing the LSB values of green, red and blue pixels using **spatial domain filtering** (*in which the value of any given pixel in the output image is determined by applying some algorithm to the values of the pixels in the neighborhood of the corresponding input pixel*) such a way that they correspond to the 8-bit binary value of the text that is to be encoded and then decoding the same.



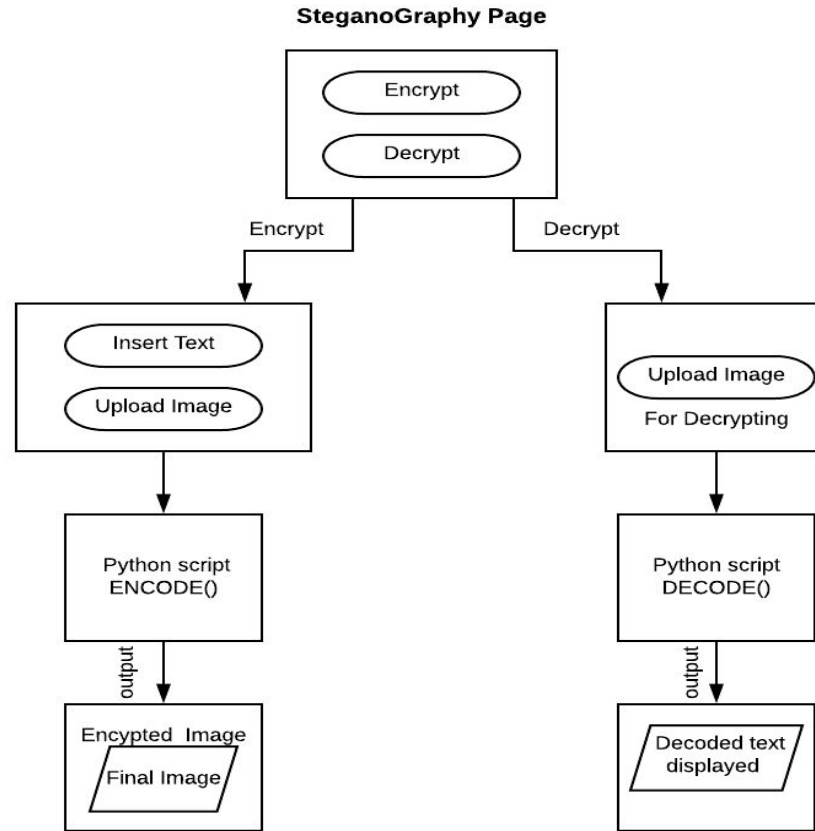
Figure 2.1 Original value of red component



Figure 2.2 Red component is replaced with binary of 65 i.e. A.

- If the user selects 'encode', they are directed to a page where they enter text. Then the text is converted into its 8-bit equivalent and is then passed as an object to flask along with an image which they wish to use to hide the text .
- The pixels of the image are expanded using the python's pillow module and converted to 8-bit strings.
- Then the 8-bit string of pixels is iterated with the 8-bit text with right most bits of the image received. That's how final image is obtained.

Similarly, for the decoding part, we first iterate over the pixels of the image uploaded and convert it into its binary representation. These binary strings obtained are checked for valid ASCII values and mapped accordingly.



WORKFLOW OF OUR PROJECT

RESULTS OBTAINED



Fig 1: shows design page for encrypting and decrypting



Fig 2: Mobile view of home page

Home Encrypt Decrypt

Enter the text here:

hey

Choose File pic.bmp

submit

Fig 3: Encryption page asking for text and cover image

your encoded image looks like this

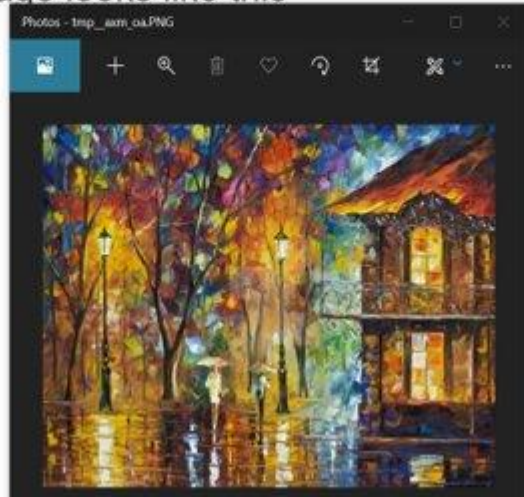


Fig 4: Successful encryption

[Home](#) [Encrypt](#) [Decrypt](#)

Please upload the file you want to decrypt here:

newpic.bmp

Fig 5: Decryption page asking for cover image

The encoded message in the provided
image is:

hey

Fig 6: Successful decryption page showing the
decrypted text

CONCLUSION

We have successfully created and implemented the model to hide a text inside an image. Steganography, is a fairly new idea, however, there are constant advancements in the field. There soon will be more efficient and advanced techniques for steganography. All the major image file formats have different methods of hiding messages, with different strong and weak points respectively.

FUTURE SCOPE

1. Detecting steganography in audio or video files
2. Steganography on the world wide web
3. Steganography in printed media

LIMITATIONS

1. If the decrypting script has been exploited or meddled with then the information that had been sent could be lost forever.
2. Huge file sizes after encrypting the messages might arise suspicion.
3. The methodology if in the wrong hands can lead to undetected illegal activities.

REFERENCES

- [1] Chandramouli, R., & Memon, N. (2001, October). Analysis of LSB based image steganography techniques. In *Proceedings 2001 International Conference on Image Processing (Cat. No. 01CH37205)* (Vol. 3, pp. 1019-1022). IEEE.
- [2] Chandramouli, R. and Memon, N., 2001, October. Analysis of LSB based image steganography techniques. In *Proceedings 2001 International Conference on Image Processing (Cat. No. 01CH37205)* (Vol. 3, pp. 1019-1022). IEEE.
- [3] Ahmad, Syukri Sayyid. "Steganography for inserting message on digital image using least significant bit and AES cryptographic algorithm." *2016 4th International Conference on Cyber and IT Service Management*. IEEE, 2016.
- [4] Ahmad, S. S. (2016, April). Steganography for inserting message on digital image using least significant bit and AES cryptographic algorithm. In *2016 4th International Conference on Cyber and IT Service Management* (pp. 1-6). IEEE.
- [5] "Image Steganography in Cryptography - GeeksforGeeks", *GeeksforGeeks*. [Online]. Available: <https://www.geeksforgeeks.org/image-steganography-in-cryptography>
- [6] R. Roy, "Hiding data in an image : Image Steganography using Python", *Medium*, 2020. [Online]. Available: <https://towardsdatascience.com/hiding-data-in-an-image-image-steganography-using-python-e491b68b1372>. [Accessed: 03- Jun- 2020]