NATIONAL INSTITUTE OF TECHNOLOGY PUDUCHERRY

(An Institution of National Importance under MHRD, Govt. of India)

KARAIKAL – 609 609

Hiding secret message using Image Steganography

Report - Network Security Project

Submitted by:-
Ghanta Adarsh - CS19B1018
Gopikrishna - CS19B1020
Moses Chandramouli - CS19B1037
Ritvik Agrawal - CS19B1045

Index

Abstract

Steganography is the practice of concealing a file, message, image, or video within another file, message, image, or video. Steganography is more discreet than cryptography when we want to send secret information. On the other hand, the hidden message is easier to extract.

Platform & Software

Google Colab and Python.
Google Colab is used as a platform to run the program and the program is written in python.

Problem statement

How can we send a message secretly to the destination? Using steganography, information can be hidden in carriers such as images, audio files, text files, videos and data transmissions.

Proposed Method

In this method, we're hiding the message using Image Steganography.

Image encoding and decoding

The basic idea here is, each pixel is made up of three basic colours (Red, Green, and Blue). We can use the parity of these colour values to store the secret message in binary format.

For 1 byte of information, we take 3 pixels which contain 9 colour values combined. In this 8 of them can be used to store 8-bit values of 1-byte information. The 9th colour can be used to indicate whether there exists the next byte of information or not.

How do we store the value in the colours? The idea is to store them in terms of the parity of the colours. If the bit value is one, we make sure the colour value is of odd parity, else we increment the colour by one to make it odd. If the bit value is zero, we make sure the colour value is of even parity else we increment it by one.

As we're incrementing the colour values at most by one, it won't affect the overall colour of the image.

Steps in encoding and decoding

Encoding

1) For each character in the message take 3 pixels. So, we get 9 colours [R1G1B1 R2G2B2 R3G3B3].
2) For each bit of binary representation of the ASCII value of the characters, do the following operation:
   -> If it is one, make sure the corresponding colour has odd parity by changing the colour to the nearest odd value.
   -> If it is zero, make sure the corresponding colour has even parity by changing the colour to the nearest even value.
3) The 9th colour will be used as a termination indicator. If we reach the end of the message, we assign it to the nearest odd value, else the nearest even value.

Decoding

1) Take a window of 3 pixels at a time.

2) Decode the character using the following method:
   -> If the parity of the colour is odd, then the bit value is one.
   -> Else it is zero.
3) Take the 9th colour value. If it has odd parity it implies we reached the end of the message else there is another character after that.

Message encryption and decryption

For encryption, password is used to generate a hash value of size 256 bits using SHA-256. This 256-bit size hash value is used as the key for the AES encryption model. Then the AES algorithm is used to encrypt the message.

For decryption, the same hash value is used to decrypt the message in the AES decryption.

Steps in encryption and decryption

Encryption

1) Password and message are taken as parameters of the encryption function.
2) The password is transformed to a 256 hash value using SHA-256 which is taken as the key for AES encryption.
3) A random block of 16-byte value is taken as the initialisation vector (IV).
4) The key and IV are used to initialise the AES encryptor object.
5) The padding amount is found and is padded with the message for perfect block size.
6) The message is then encrypted using an AES object to get a ciphered message. IV is added to the beginning of this ciphered message.
7) The final message is encoded in base 64 standard and is sent to the encoding function.

Decryption

1) Password and ciphered message are taken as parameters for the decryption function.
2) The ciphered message is converted from Base 64 format to byte format.
3) The password is transformed to a 256 hash value using SHA-256 which is taken as the key for AES decryption.
4) The first 16 bytes are taken as IV which is used along with the key to initialise the AES decryptor object.
5) Then the ciphered message is passed to the decryptor object to decrypt the message.
6) Padding is removed from the deciphered message and the final decrypted message is returned.

Modules used

PIL

Python Imaging Library is a free and open-source additional library for the Python programming language that adds support for opening, manipulating, and saving many different image file formats.

Crypto.Cipher

The Crypto.Cipher package contains algorithms for protecting the confidentiality of data. It has the AES module inside it.

Crypto.Hash

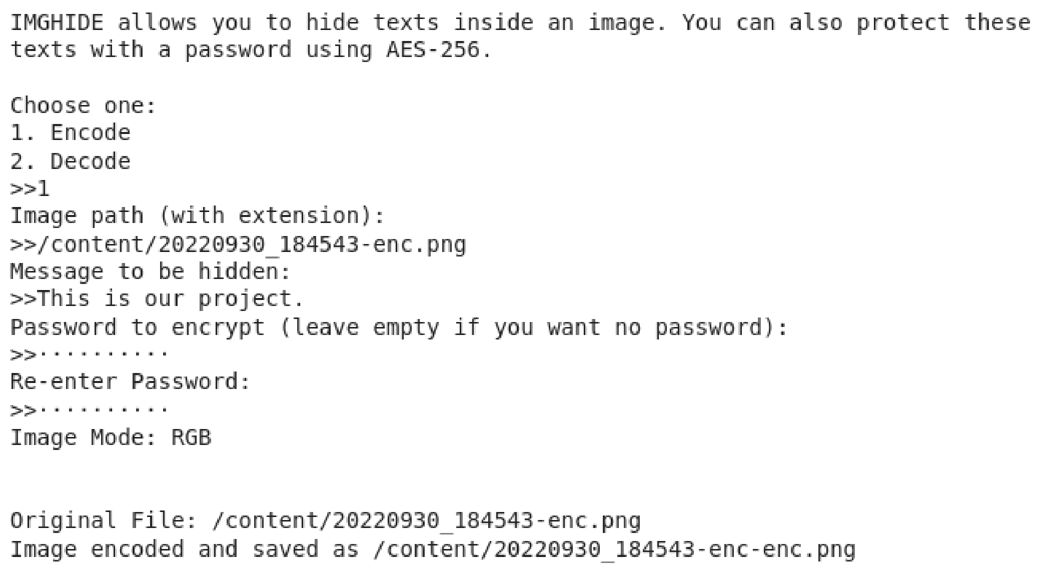
It contains the SHA-256 module which takes arbitrary binary strings as input and produces a random-like fixed-length output (called digest or hash value).

Crypto.Random

It returns a random byte string of length N.

Output

Encoding

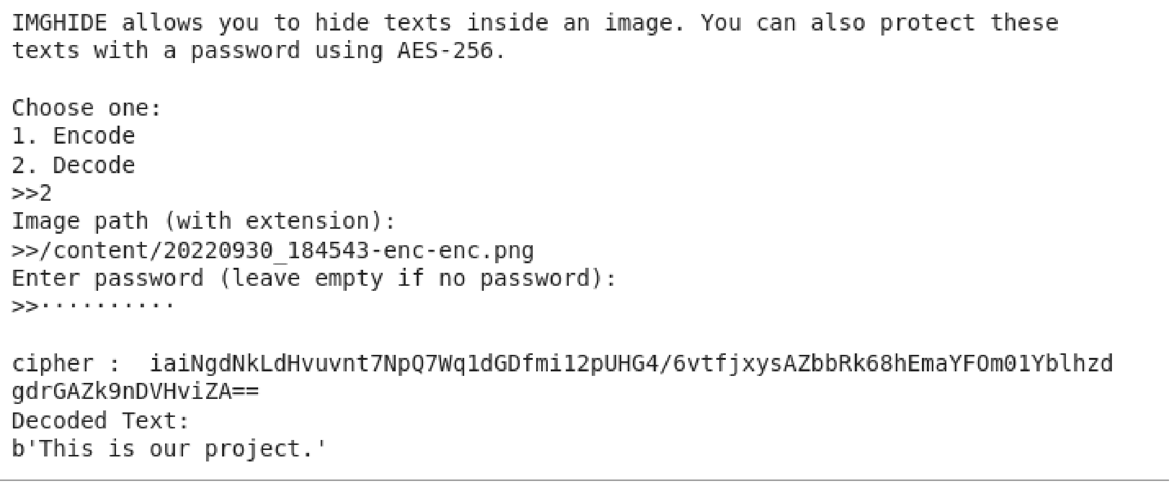


```
IMGHIDE allows you to hide texts inside an image. You can also protect these
texts with a password using AES-256.

Choose one:
1. Encode
2. Decode
>>1
Image path (with extension):
>>/content/20220930_184543-enc.png
Message to be hidden:
>>This is our project.
Password to encrypt (leave empty if you want no password):
>>··········
Re-enter Password:
>>··········
Image Mode: RGB


Original File: /content/20220930_184543-enc.png
Image encoded and saved as /content/20220930_184543-enc-enc.png
```

Decoding:



```
IMGHIDE allows you to hide texts inside an image. You can also protect these
texts with a password using AES-256.

Choose one:
1. Encode
2. Decode
>>2
Image path (with extension):
>>/content/20220930_184543-enc-enc.png
Enter password (leave empty if no password):
>>··········

cipher :  iaiNgdNkLdHvuvnt7NpQ7Wq1dGDfmi12pUHG4/6vtfjxysAZbbRk68hEmaYFOm01Yblhzd
gdrGAZk9nDVHviZA==
Decoded Text:
b'This is our project.'
```

Conclusion

Steganography is not intended to replace cryptography but rather to supplement it. If a message is encrypted and hidden with a steganographic method it provides an additional layer of protection and reduces the chance of the hidden message being detected.

Steganography goes well beyond simply hiding text information in an image. Steganography applies not only to digital images but to other media as well, such as audio files, communication channels, and other text and binary files.

Reference

https://www.mygreatlearning.com/blog/image-steganography-explained/
https://www.simplilearn.com/what-is-steganography-article