Comparative Analysis Of Various Steganographic Techniques

Prachi Agrawal[1], Oindreela Bhowmick[2], Yash Prince[3]

[1]Department of Applied Mathematics, Delhi Technological University, India
[2]Department of Applied Mathematics, Delhi Technological University, India
[3]Department of Applied Mathematics, Delhi Technological University, India

Article History: Received: 11 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 10 May 2021

Abstract: In the era where basic mode of communication is happening on a digital space or as we say on online int"rnet based spaces and a concern towards privacy or secrecy is bound to enter the needs of users. That section is where this Paper particularly aims. To provide a detailed comparative analysis between different techniques present out to carry out steganography and Cryptography. These two techniques are independently capable of creating a secured secret shareable communication means. But this paper will aim on the different implementations of steganographic techniques and give detailed research outputs on their real life applications.

Keywords: Convolutional Neural Networks, Genetic Algorithm, Hamming Code, Pixel Permutation, Steganography

1. Introduction

I.1 Definition

The word **Steganography** comes from '**grayfia**' meaning 'writing' and '**stegos**' which is 'to cover' from Greek language thus translated to '**hidden writing**' or '**covered writing**'Steganography embeds secret data into an image, video or an audio or a computer file. This method is employed to guard the hidden sensitive secret data or information from unauthorized people or hackers.

Steganography tends to have an advantage over cryptography as a secret message which is intended to keep secured isn't visible in plain sight and doesn't pull attention to itself.

Steganography can include various methods like the concealment of data or like within media files with steganographic coding inside a transport layer like an image file or a document file

Media files due to their enormous sizes are considered practically ideal for using techniques like steganographic transmission.

Steganography is applicable to, but not limited to, the subsequent areas :

- Confidential communication and secret data storing
- Protection of knowledge alteration
- Access system for digital content distribution
- Media Database systems

The area differs in what features of the steganography are used in each system.

Cryptography is that practice which is used to protect the information of the secret message whereas steganography is worried both about concealing the actual minor detailed facts that a message with secret hidden is being sent

1.2 Types of Steganography

Depending on the type of cover object used in the technique application in which the secret is embedded, we can divide steganography into different types : -

- **Text Steganography**

It is the technique to hide data within text files. It can involve changes like formatting of current text or changing words or when we use context free grammar to get readable texts

- **Image Steganography**

When an image is used as a cover object it is called image steganography. They are widely preferred because of their large number of pixels present in digital representation.

- **Video Steganography**

We can hide a big amount of data within a digital video format as this size number can hold a large amount of secret message

- **Audio Steganography**

When an audio signal is embedded with a secret message which then alters the binary composite sequence of the audio file is called audio steganography.

- **Network Steganography**

When information is embedded in data transmission in network-controlled protocol.

2. IMPLEMENTATIONS

II.1 Text in Image - LSB algorithm

II.1.1 L S B overview- It is the most common strategy which is preferred for steganography. And if we choose to go additional by utilizing pixel data of an image then steganography can be a prominent tool used in steganography. It replaces each bit of the double repetitive content piece with singular one piece of each pixel within the pictures which was first used. When the cover record is comparatively longer than the message then this technique works the best or if the image is grayscale it takes a twenty four 24 bit picture and converts each bit to 3 bit which is then encoded in each and every pixel. Example we can replace the last least significant bit and use it to cover data or information for every colour's byte without actually disturbing a significant amount of change which isn't visible to a naked eye.

II.1.2 Procedure of encoding

Each and every byte of knowledge is then converted to its respective eight bit code when used values from ASCII . Left to right is the way how pixels are read during a group of three containing a total set of nine values binary data is normally stored in the first eight values the value when converted comes out odd irrespective of presence of one or zero[6] For eg let's take the message which we are to hide is ' Hiiii ' and as the message we have is of three bytes hence pixels we will be requiring to encode the info is 3x3 = 9. Now if we take an image 4X 3 with complete twelve pixels which are enough to encode the present data :

[(27, 64, 164), (248, 244, 194), (174, 246, 250), (149, 95, 232),(188, 156, 169), (71, 167, 127), (132, 173, 97), (113, 69, 206),(255, 29, 213), (53, 153, 220), (246, 225, 229), (142, 82, 175)]

The value using ASCII code for 'H' is 72 which has an equivalence in binary equal to 01001000. Now if we take the first set of three pixels which are 27 64 164 and 248 244 194 and 174 246 250 and then use them to encode information. Now we can change the value of the pixel and make it odd for 1 or even if it is 0. Then results in modification of pixels which are now 26 63 164 and 248 243 194 and 174 246 250 and now Since we've to encode more information therefore we should keep the last value even. Similarly ‘i’ will be encoded during processing of the current image and the resultant image will be like

[(26, 63, 164), (248, 243, 194), (174, 246, 250), (148, 95, 231),(188, 155, 168), (70, 167, 126), (132, 173, 97), (112, 69, 206),(254, 29, 213), (53, 153, 220), (246, 225, 229), (142, 82, 175)]

II.1.3 Decoding the information

To decode a set of three pixels are processing reads till the time we encounter the last value to be odd which implies we have reached the end of the message. Every set of three pixels contains one binary bit of information which then can be retrieved by the identical logic used for encoding when the binary value of bit is 1 then it means the value is odd or else zer

II.2 Image inside Image Steganography

II.2.1 Algorithm used : LEAST SIGNIFICANT BIT

Pixels are little individual elements of a picture. So each pixel may be a sample out of the original image which can mean that there are samples comparatively more which provide representations which are more accurate of the fir The intensity of every pixel when discovered is variable. Representations by 3 or 4 component intensities are used in color imaging systems like green blue and cyan black yellow or magenta. We are here using RGB model which contains 3 channels it works as an additive to color model during which R, G and B

lights are superimposed in numerous ways to breed a wide range of colours. It is named after the three primary initials of colors. Mainly this model is used for representation sensing and displaying of images in electronic systems like computers and televisions though it's also been employed in conventional photography.

So as we notice here that every pixel from the chosen image consists of three values R G B with values of 8 bit and ranging between 0 to 255.

As we now notice to see that the image which is just above every pixel we've three values which might be represented in code for computer language. We tend to have more number of significant bits and less significant bits also when we start working with codes in binary

The bit on the leftmost side is the one which is the most vital bit. If we alter the leftmost bit it'll have an oversized impact on the ultimate value for instance if we modify the bit from the leftmost side to 0 from 1 then it changes 11111111 to 01111111 which affects the value to change from 255 to 127. And in the other case if we take the btit from the rightmost side that is that the diminished bit If we modify the rightmost bit it's tend to have comparatively insignificant impact on the resultant value as an example when we alter bit from the rightmost side to 0 from 1 itresults in 11111111 to 11111110 resulting in the value to change from 255 to 254 as we can notice that when we alter the bit from the rightmost then it will change just one in every range of 256.

Summarizing the value of each and every is 8 bit meaning it can only store values of 8 binary and also bits from the rightmost side are lower. So if we modify the rightmost bits it'll have a low noticeable impact on the ultimate image. This is often used as the steganography method key to cover a picture inside a different picture. Change the lesser bits from a picture and include the foremost significant bits from the opposite image.

II.2.2 Image Hiding :

1. The image that is secret should be of the same size as the picture that will conceal it to hide an image within another.
2. From all the rows and columns we need to create two loops from the images.
3. Therefore we get RGB as binary values from both images and can use the int to bin function to convert from decimal to binary.
4. We combine the most important bits from the 1st image with the most important bits from the 2nd image. We use the function __merge rgb which takes four most important bits from every image but it can be altered.Using a lesser number of bits from the secret image would result in a poor image recovery quality.
5. Finally using the __bin to int process we get the binary to decimal value and give it a new pixel location in the resulting image.

So we get an image hidden inside another image.

II.2.3 Image Revealing :

1. We should know number of bits which were used to mask the picture in order to show an image We use four bits in this scenario
2. First we should go through the pixels of the image so that we can create two loops
3. Using the method 'int to bin' we take each RGB channel from the current pixel as a binary value
4. Then to get a new eight bit value we establish a new RGB value by adding just the right four bits of the current pixel with 0 values.
5. Finally in the new picture we get the binary into a decimal value and give it to the current pixel
6. When the secret image size is smaller than the image that hides it, the built technique has only one last step to erase the black borders and we can get an image from the other.

II.3 Steganography using Genetic Algorithm

Genetic Algorithm Genstego is a picture steganography package supported by deaIt encodes and decodes secret messages that are in the form of images into groups of images. The idea behind this algorithm is when we when we consider using genetic algorithm to model steganography as an optimization problem or as a probe[1]. Optimal stochastic solutions can be found using these. Therefore making the steganalysis more complex Basic image steganography. We can represent an image in a matrix ranging between zero to two hundred and fifty five. It is resembling eight bits that is amongst the common basic algorithms of image steganography which includes LSB.We need to change lsb of every pixel of host image with bits of key image that is eight pixel of host id

1 "Genstego: Image Steganography Based on a Genetic Algorithm." Genstego: Image Steganography Based on a Genetic Algorithm. Accessed 15 Apr. 2021.

needed to hide one secret pixel.For hiding the image of square size sixteen,we need at least square size forty-six. So to get the key image,we need to retrieve host pixels and make a key image.

This algorithm is not perfect as anyone can decode the message. Also Least Significant Bit occupies more space in the host image so we will be optimizing the technique and use it in steganography.

II.3.2 Details of Genetic Coding

In next table we find how a following chromosome is composed of 7 genes :

A full chromosome has a length of 26 bits

Direction
Depicts the direction in which the cover image is scanned so that the bits from the key image can be embedded in the cover.

X-offset and Y-offset
Depicts the column and row, that is, the location from where the actual scan begins. It contains a length of one byte because we have used cover images of 256x256 width and height but that can be easily modified.

Bit-Planes
This has four bits and represents which bits of the cover image are manipulated to insert the key bits. For example, if bit-Planes are 0001 it's best to use LSB but if the bit-plane is 0101 the 3rd and fewer significant bits are used.

SB-Pole
If the bit is one, the 1's complement is involved in the key bit sequence. For example, the bit sequence 1001 should be changed to 0110.

SB-Dire
If the worth is 1 the one's complement is applied. For example the 0001 should be changed to 1000.

BP-Dire
When the bit is 1 the selected bit-planes are hidden inside the 4 LSB of the pixel in consideration. In the case it is zero, the 4 MSB (Most Significant Bits) are used.

Fitness function : PSNR (Peak Signal to Noise Ratio)
The ratio between the maximum potential power of an image and the power of corrupting noise that affects the quality of its representation is the peak signal-to-noise ratio (PSNR). It compares the image to an ideal clean image with the greatest possible power to estimate the PSNR of an image.It is measured by Decibels
It calculates the standard of stego images.[12]

$$PSNR = 10 \cdot log_{10}(\frac{MAX_I^2}{MSE})$$

$$MSE = \frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}||I(k,j) - K(i,j)||^2$$

For getting a less noisy host image,we need to maximize this function as greater the value,lesser noisy the image is.

II.3.3 Characteristics of this algorithm
This algorithm is very straightforward and to put it in a nutshell :
1. The individuals are assessed and their initial values are set randomly.
2. Then, a replacement population of people are selected by tournament.
3. The individuals of the generated population are mates that implement a two-point crossover. [10]

And currently generated individuals are muted (mutation concept) by flipping of random genes with a probability by swapping random genes of the not operand chromosomes.

II.3.4 Hiding and revealing a secret message
In order to accomplish the process of hiding the secret image using any of the cover and the bit sequence are flattened secret images.For the cover image, the sequence is flattened with the help of Direction and ranging

from y and x offset.[10][11][12] Then the bits of key are altered reckoning on SB-Dir and BP-Pole. So if the number of bits available within the host image is smaller than the sequence of secret bits, then it is ready at 0 fitness. Bits in different cases are being embedded into the following configuration which can be calculated by bp-Dir and bit planes. We need to get the key raw bits with the Bit-Planes and BP-Dir and make it into readable form with the help of SB-Dir.

II.4 Video Steganography using Hamming code

This algorithm demands a video to then get converted into different frames and then altering the position of the pixel. Conversion of the message must be done to one dimension either XOR and hamming code before actually transmitting it XOR and Hamming code are applied to the current Message. the message which is encoded and embedded with video frames which are scrambled. Frames which are being sent to a receiver after reconstruction then a receiver can only extract the information which is using the identical key because the sender side The receiver can extract information using the identical key because the sender side. The receiver has got to use XOR or hamming code used to disassemble the video and reconstruct the right intended message

2.4.1 ALGORITHM : (7, 4) Hamming Code

The Hamming code is one of the major well-known block code methods that is known to execute both error detection and correction on a given block of message. The algorithm of hamming code is that it is coded by adding some extra padding with only repetition of least amount which is named the codeword, bits of length n. [2]The section added contains information parity of (n-k) length bits where length of message is k that is expected to be coded. The Hamming codes are linear codes in order .It consists of two matrices : 1. parity-check matrix H and 2. generator matrix G, which they have for both encoder and decoder. For the encoding part, each message M, consisting of four bits, will be multiplied to G then we get a modulo of two applied; the results achieved is a codeword X of value 7-bits which is currently to be sent through a loud channel.

On the decoding part, for the objective of checking the encoded message of 7-bits R (data + parity) are received, then they are multiplied by the transpose of the parity-check matrix, and taking modulo of two again.

The result is a syndrome vector Z (z1, z2, z3) of three bits, which should be all zeroes if it is an error-free message. Otherwise, any change inside the message during communication will result in flipping one or more bits of the message; then it will require a miscalculation correction technique.

II.4.2 Designation

The following points are implemented in MATLAB :

Sender's Side:

- Convert the stream of the video into a different number of frames.
- Divide each frame into the components of U V and Y
- Using a special key alters the positions of these components.
- The message that needs to be go to 1 D array should be converted and then alter the place of the message by a key
- Using the Hamming 7 4 encoder encrypt every four bits of message.
- The product of encoded data consisting of seven bits that is four bits of message + three bits of parity is XORed with the seven bits of any value.
- The pixels are back to their original positions and the video is built again.

Receiver's Side:

- Video is converted into frames
- Separate the components V Y and U for each frame
- Change the location of all pixel values in the 3 components U V and Y with the special key used in the process of embed
- To retrieve the data that is encoded from UVY,we use the exact key used by the sender side and XOR components with any number.
- Four bits of message should be decoded from the Hamming decoder
- Relocate the whole message in the original order again.

2 "(PDF) Video Steganography - ResearchGate." 26 Jan. 2018, (PDF) Video Steganography. Accessed 20 Apr. 2021.

- Convert the message sequence to two dimensions.

II.5 Steganography using Cryptography and Deep Neural Networks

Steganography has usually been integrated with cryptography and neural networks separately in earlier works but this algorithm combines both to form a secure medium to transfer private information. It is an image steganography technique which embeds a secret image inside an unsuspecting cover image of the same or larger size. It becomes an effective technique when the simple cryptographic method used here is combined with convolutional neural networks.

Architecture :

As shown in the figure, the network architecture consists of 4 components in 2 blocks. All of these are trained as a single network. The working of all the components is described as follows :

- The secret image S is passed through the **Prep Layer**, where it undergoes various transformations to produce the Encrypted Secret Image (S_e), The most important intention of this layer is to finally plant the secret image into the encrypted image so as to steer clear of any disclosure of the secret message.
- A cover image along with the output of the Prep Layer forms the input of **Hiding Layer.** The output of this layer is the stego image that is to be passed on to the receiver. Prep layer and Hiding layer form the Sender Block.
- The third Layer, **Reveal Layer**, constructs the encrypted image again. This layer takes in the Stego image as input and separates the cover image from it to generate the encrypted secret image.
- The fourth layer, the **Decrypt Layer**, receives the Reveal Layer's output and decrypts it. Finally, the hidden picture will be revealed. The third and fourth layers are combined to make the receiver block.

Optimizer : The optimizer's job is to improve the weight parameters in order to reduce the loss function. It assists the loss mechanism in locating its global minima. The ADAM - Adaptive Moment Estimation optimizer is used in the model, which measures a different learning rate for each parameter. It is computationally efficient and has relatively little memory, making it ideal for this model.

Image Hiding

Since text is more tough to process tha images, they are commonly used in communication. According to Kumar, the user can interact on a compromised network, in which case the security of the data becomes critical.The given 3 concepts are at the heart of image hiding process and algorithms:

• Pixel Permutation - rearranging the pixels
• Pixel Substitution - changing the value of pixel
• Visual Modification

For encryption methods,an image histogram is the key feature taken into account . The histogram shows the frequency variation and gives information on how often each and every pixel value occurs.The process based on pixel value transformations, such as Advanced Encryption System, tries to construct a stable image histogram that secures the image from the attack of plain text while requiring 0 loss when image decryption is performed. This isn't helpful in this situation since the neural network is being used to conceal

photos in other photos and is determined by the redundant part in the cover images, which gives the sum of data that has been hidden. Similarly, Elliptic Curve Cryptography is a form of cryptography that uses elliptic curves to encrypt data.

In our case, this explanation cannot be used.

Above discussed methods fail due to information loss in neural networks. As a result, to produce the identical histogram ,a process based on scrambling data with the identical pixel value but in a different position can be used . The scrambling keeps the histogram the same but reduces the similarity between pixels. [15]

The technique of shuffling frames of pixels is used in this model.

The correlation decreases as the counting of frames increases, resulting in an appropriate hiding layer. The order of encryption is determined by counting of frames . When the encryption order is increased, it is discovered that one is not able to read the meaning of the picture while looking.

What If the Original Cover Image Is Available?

If the original cover image can somehow be obtained through other sources, then it is possible to retrieve some information about the secret image without even using the decoder by comparing the original cover image with the stego image.

However data can be exposed without the layer of encryption by calculating the difference between the initial and restored cover images. Though the secret image is not visible after obtaining the difference, it is visible after enhancing the image features. Since transforming it to grayscale, we observe that the hidden image and the original image have a striking similarity.

If we use the first layer of encryption, the key is difficult to acquire because after improving and transforming the residual image into grayscale, we only see boxes and the residual image is unintelligible (see Figure). As a result, the device is stable, and extracting the characteristics of the hidden picture is almost impossible.

3. Our results

III.1 Text in Image - LSB algorithm

cover.png

stego.png

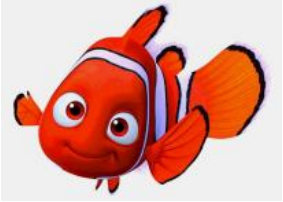
```
G:\DTU\Projects\B.tech Project\LSB>pip install pillow
Requirement already satisfied: pillow in c:\python38\lib\site-packages (8.0.1)

G:\DTU\Projects\B.tech Project\LSB>test.py
:: Welcome to Steganography ::
1. Encode
2. Decode
1
Enter image name(with extension) : babyyoda.png
Enter data to be encoded : hello
Enter the name of new image(with extension) : stego.png

G:\DTU\Projects\B.tech Project\LSB>test.py
:: Welcome to Steganography ::
1. Encode
2. Decode
2
Enter image name(with extension) : stego.png
Decoded Word : hello

G:\DTU\Projects\B.tech Project\LSB>
```

III.2 Image inside Image: Least Significant Bit

cover.png

secret.png

stego.png



extracted_image.png

References

1. "Steganography | Definition of Steganography by Merriam-Webster." Steganography | Definition of Steganography by. Accessed 14 Apr. 2021.
2. "Steganography - Wikipedia." Steganography. Accessed 14 Apr. 2021

3. "Image Steganography in Cryptography - GeeksforGeeks." 9 Aug. 2019, Image Steganography in Cryptography. Accessed 14 Apr. 2021.
4. "Applications of Steganography." 24 Aug. 2018, Applications of Steganography. Accessed 14 Apr. 2021.
5. "Steganography Tutorial | A Complete Guide For Beginners | Edureka." 25 Nov. 2020, Steganography Tutorial | A Complete Guide For Beginners. Accessed 15 Apr. 2021.
6. "(PDF) A Steganography Technique for Hiding ... - ResearchGate." (PDF) A Steganography Technique for Hiding Image in an Image using LSB Method for 24 Bit Color Image. Accessed 14 Apr. 2021.
7. "Image based Steganography using Python - GeeksforGeeks." 20 Aug. 2020, Image based Steganography using Python. Accessed 15 Apr. 2021.
8. A New Method of Image Steganography Using 7th Bit of a ... - Hindawi." A New Method of Image Steganography Using 7th Bit of a Pixel as Indicator by Introducing the Successive Temporary Pixel in the Gray Scale Image. Accessed 15 Apr. 2021.
9. "Steganography: Hiding an image inside another | by Kelvin Salton ...." 17 Mar. 2018, Steganography: Hiding an image inside another | by Kelvin Salton do Prado. Accessed 15 Apr. 2021.
10. "Introduction to Genetic Algorithms — Including Example Code | by ...." Introduction to Genetic Algorithms — Including Example Code | by Vijini Mallawaarachchi. Accessed 15 Apr. 2021.
11. "Genstego: Image Steganography Based on a Genetic Algorithm." Genstego: Image Steganography Based on a Genetic Algorithm. Accessed 15 Apr. 2021.
12. "Genetic Algorithms - GeeksforGeeks." 23 Aug. 2018, Genetic Algorithms. Accessed 15 Apr. 2021.
13. "(PDF) Video Steganography - ResearchGate." 26 Jan. 2018, (PDF) Video Steganography. Accessed 20 Apr. 2021.
14. "(PDF) A Highly Secure Video Steganography using ... - ResearchGate." (PDF) A Highly Secure Video Steganography using Hamming Code (7, 4). Accessed 20 Apr. 2021.
15. "Hiding Data in Images Using Cryptography and Deep Neural Network." 22 Dec. 2019, [1912.10413] Hiding Data in Images Using Cryptography and Deep Neural Network. Accessed 20 Apr. 2021.