A New Approach to Stegnography

Soumik Mukherjee, Moumita Deb, Pratik Kumar Agarwal, and Abhinava Roy

RCC Institute of Information Technology

Abstract. Steganography hides the fact that a message is being sent. It provides security. In this world full of intruders, security of information is the most vital concern. In this paper, we present a Steganography algorithm. Our approach is based on hiding message in an image file by selecting pixels using some mathematical formulation and replacing the last few bits of the pixel. The bit selection is also not obvious; the selection is decided again by some calculation. In our algorithm we try to ensure that there is minimum amount of distortion from the original image. We also use a symmetric key to ensure security of the message. To the best of our knowledge this approach can be used to hide data in an image file with great security and with a very small chance of detection.

Keywords: Steganalysis, Carrier File, Stego-Medium, Redundant bits, Payload, Steganographic Algorithm.

1 Introduction

Steganography[1] is the art and science of writing hidden messages in such a way that no one, apart from the sender and intended recipient, suspects the existence of the message, a form of security through obscurity. Generally Steganography has been derived from the Greek words Stegos (covered) [1] and Grafia (writing) [1].The main objective of Steganography is to communicate securely in such a way that true message is not visible to the observer, that is the unwanted parties should not be able to distinguish in any sense between the covered image (image not containing any secret message) and stego image (modified cover image that containing secret message). However Steganography differs from Cryptography[6] in the sense that the former focuses on concealing the existence of a message while the latter focuses on concealing the contents of a message. However Steganography differs[7] from Cryptography in a sense that former hides without altering the bits while the latter alter the bits without hiding. There is another difference between Steganography and Cryptography, Steganography is the process of embedding information within other seemingly harmless information in such a way that no one but the intended can retrieve it while cryptography is the process of transforming information into other unintelligible such that no one but the intended recipient can able to retrieve it.

Generally, Steganography was thrown into light by first Greek Historian Herodotus (485-525 B.C) [4]who shaved up the head of the messenger and then wrote the message into his scalp and then waited for the hair to regrow.The messenger apparently carrying nothing contentious, could travel freely. Arriving at his

N. Meghanathan et al. (Eds.): CCSIT 2012, Part I, LNICST 84, pp. 238–247, 2012.

destination, he shaved his head and pointed it at the recipient Steganography works as given below in the diagram.

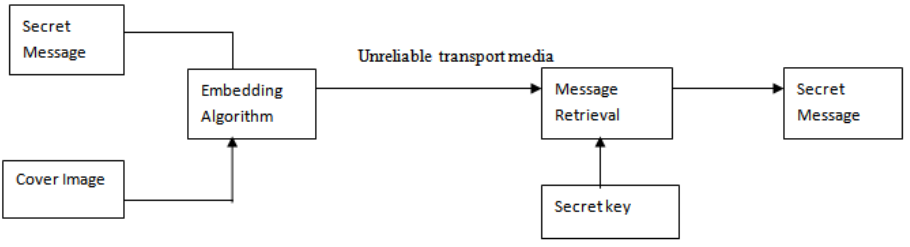

Fig. 1. Block diagram of Steganography process

To hide a secret message in an image without changing its visible properties, the cover image can be altered in the "noisy" regions with many colour variations, so as to reduce the attention drawn towards the altered location. The most common methods of alteration include changing, masking, filtering and transformation of the bits is carried out at the least significant bit (LSB), developed by Chandramouli and others [2].Later came the construction of an algorithm for detecting LSB steganography. There is continuous changes[5] in the cutting edge world of steganography and the large amount of data involved, so steganalysists have tried to design algorithms to improve the hiding capacity of images and lowering the detecting rates.

If we use a 24-bit image, a bit from each red, green and blue component can be used. So for an image of 500x400 pixel image can contain a total of 1600000 bits and a secret message of 600000 bits (75000 bytes).But using just 3 bits out of every 24 bits is waste of huge amount of space. So the main objective of the current work is to insert the message in more than one bit in each byte of the pixels of the cover image and still get a result similar to the LSB substitution (imperceptible to the naked eye). This objective is achieved by using a new steganography algorithm by hiding a large amount of data in bitmap images by using maximum number of bits per byte in each pixel. We can discuss two types of attacks to be sure that our process for embedding data is working efficiently. The first attack concerns to work against visual attacks,to make use of the ability of humans to unclearly discern between noise and visual patterns, and the second attack concerns to work against statistical attacks to make it much difficult to automate.

2 Our Idea Behind Steganography

Our main aim of this algorithm is to pass a secret message (within an image) with minimum distortion of the image so as to increase security and to reduce the chances of detection. But before going onto the main algorithm part, let us overlook some of the important concepts :-

a) Segmentation of the cover image: -We segment the image into blocks of 24 bytes each. We use uniform segmentation [3] because it saves space and there is less amount of wastage in this kind of segmentation.(Fig.2).

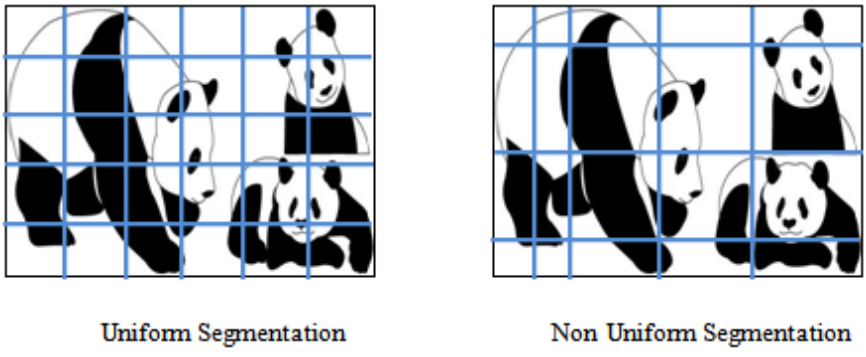

Fig. 2. Different way of segmentation of cover image

b) Pixel Selection: - This is another layer of security in the algorithm; we perform almost random selection of pixels based on their colour component and a secret key.
c) Bit Infection: -This is third layer of security in the algorithm; we insert three bits in place of a single bit but instead of adding more information we add more security. For a'0' we inject '101' and for '1' we inject '010'.Moreover if a specific pattern of bit is absent we spread these three bits into three different bytes.

2.1 Description of Our Algorithm

The present algorithm has two parts: one is the message hiding part at the receiver's end and another is the message extraction part at the receiver's end. These parts are developed and implemented for the following reasons stated below:-

1. This algorithm reduces the chances of statistical detection.
2. The algorithm must provide robustness against a variety of image manipulation attacks.
3. The stego-image must not have any distortion.
4. The algorithm must not sacrifice security to achieve extra capacity for image storing.

The first part of the algorithm is to store the message in the cover image; it is achieved by the following steps:

- Accept an image from the sender, to be used as a cover image.
- Accept a secret two digit key from the sender.
- Embed the key in the cover image.

- Perform the required calculations as described in the algorithm (section 3 of this document).
- Take the secret message from the sender and embed it into the cover image according to the procedure stated in the section 3.
- Save the image.

While the second part of the algorithm is retrieve the message from the received image. It can be achieved by following steps:

- Accept the image received by the receiver.
- Accept the secret key from the receiver.
- Check whether the key entered by the receiver is the same as the encrypting key or not.
- If the keys match extract the message.

3 Algorithm

Consider the image(I) and convert it to its corresponding binary format bin[ ].
1. Calculate prominent color component of I, i.e. if prominent color is red or blue or green then comp=0 or 1 or 2.
2. Calculate the most frequent color byte of the color component (comp) ; in max.
3. Accept secret key and convert it into its binary format in key[ ].
4. Calculate the block increment value in incr ,
   incr = (Number of bytes in I) % Integer(key[ ])
   if incr > 11
   incr = incr % 11
5. Store the pattern '00110011001100011' in the image
   for denoting the presence of a message in the image.
   Consider bytes bi[ ], i = 1 to 8;
   if(i % 2 = =0)
   Replace (bi[6-7] ,11)
   else
   Replace (bi[6-7] ,00)
6. For the next 8 bytes consider each byte as bk[ ],
   vary i = 0 to 7
   Replace (bk[7] , key[i])
7. Consider the message in its binary format as message[ ] and its length in binary format as len[ ].
   Now for the next 8 bytes consider each byte as bk[ ],
   vary i = 0 to 7
   Replace (bk[7] , len[i])
8. Now consider 8x3 byte block cover from next sets of
   bytes ,each block denoted by Ni . Repeating the set of processes based on i,
   Where i = i +24 + (incr * 3) [incrementing and repeating till the message length number of times] Considering Ni,
   8.1 Search for the presence of max in the prominent

color components (comp) in Ni.

8.2 if max is present , replace the last bit of the first byte and the last bit of the last byte of the block with 1,consider the max byte as bk[] ,

if(message[j] = =0)

Replace(bk[5-7] , 101) [type-I byte]

And check the previous existence of a byte of type –II, if present then flip the last bit and save it.

else

Replace(bk[5-7] , 010) [type-II byte]

And check the previous existence of a byte of type –I, if present then flip the last bit and save it.

else if max not present, replace the last bit of the first byte and the last bit of the last byte of the block with 0.

Find byte closest to max, say bx[ ]

if(message[j] = =0)

Replace(bx-1[7] , 1), Replace(bx[7] , 0),

Replace(bx+1[7] , 1),

else

Replace(bx-1[7] , 0), Replace(bx[7] , 1),

Replace(bx+1[7] , 0)

9. Save the new image .

Fig. 3. A block of the image in binary format(with max)

3.1 Storing of Bits in an Image

Let us consider the prominent color component being BLUE and the max = 11111111.Following our algorithm for the above example, the max byte is searched and found within the block.Now the last bit of the first byte and the last bit of the last byte of the block is changed accordingly (Fig. 4).

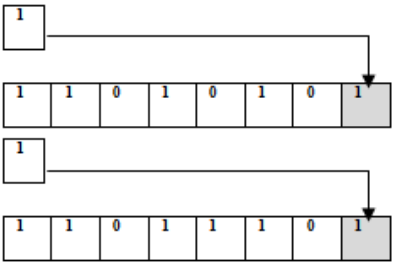

Fig. 4. Bit substitution of the first and last byte

Now if the message bit is 0, then we replace the 3byte of our block. (Fig. 5).

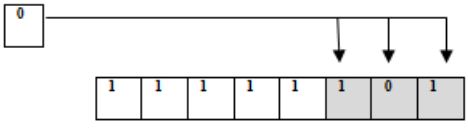

Fig. 5. Substituting the last 3bits as per the bit representation format

Now considering a situation when a block does not contain the max =11111111 byte in its color component (blue here) in Fig. 6.

Fig. 6. A block of the image in binary format(without max)

Following our algorithm for the above example, the max byte is searched and it is not found within the block.Now the last bit of the first byte and the last bit of the last byte of the block is changed accordingly (Fig. 7).

Fig. 7. Bit substitution of the first and last byte

Now if the message bit is 1, then we replace the last bit of the 12th byte of our block (Fig.8) with the message bit i.e. 1 and the previous byte or the 11th byte is transformed accordingly (Fig. 9) by replacing the last bit of the byte with 0. Moreover the 13th byte or the byte after the byte under consideration is modified by replacing the last bit of the block with 0(Fig. 10).

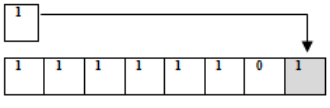

Fig. 8. Last bit substitution of the 12th byte

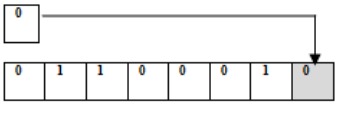

Fig. 9. Last bit substitution of the 11th byte

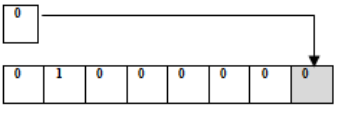

Fig. 10. Last bit substitution of the 13th byte

3.2 Retrieval of the Message

Consider the image(I) and convert it to its corresponding binary format bin[ ].

1. Calculate prominent color component of I, i.e. if prominent color is red or blue or green then comp=0 or 1 or 2.
2. Calculate the most frequent color byte of the color component (comp) ; in max.
3. Accept secret key and convert it into its binary format in key[ ].
4. Check the pattern '0011001100110011' in the image

for denoting the presence of a message in the image.
Consider bytes bi[ ], i = 1 to 8;
Check if pattern exists considering the last two bits of the bytes all together, if present then proceed
Else , notify the user about the image being void of any message.

5.For the next 8 bytes consider each byte as bk[ ],
vary i = 0 to 7
Check(bk[7] , key[i])
If found correct then proceed , else allow two more chances to the user to enter the correct secret key.

6. Calculate the block increment value in incr ,
incr = (Number of bytes in I) % Integer(key[ ])
if incr > 11
incr = incr % 11

7. Extract the length of the message from the image data in the binary format in len[] ;
vary i = 0 to 7
Extract(bk[7] , len[i])

8. Now consider 8x3 byte block cover from next sets of bytes ,each block denoted by Ni repeating the set of processes based on i,
Where i = i +24 + (incr * 3)[incrementing and repeating till the message length number of times]
Considering Ni ,

8.1 Check if last bit of the first byte of Ni block is '1' or the last bit of the last byte is '1'or not. If it's '1' then consider the max byte as bk[] ,
Search for type-I byte (last three bits of bk[] replaced by '101') or
type –II(last three bits of bk[] replaced by '010') byte in the prominent color components (comp) in Ni

8.2 if type-I byte is present , store message[ ] bit as '0'
Or if type-II byte present ,store message[ ] bit as '1'
else Notify the user of the block data being tampered and Search for the byte nearest to the max byte in bx[ ]
Search for the two bit pattern within the last 3 bits of the byte that corresponds to a message bit representation and extract the message bit accordingly.

8.3 If last bit of the first byte or the last bit of the last byte of Ni block is '0'.
Search for the two bit pattern within the last 3 bits of the byte that corresponds to a message bit representation and extract the message bit accordingly.

9. Display the extracted message.

4 Results and Discussion

Specifications of a good Steganography algorithm depend on the following :

- Comparison between the present work and the previous work.
- Human vision scale (to avoid visual attack), message can be embedded in a cover image in such a manner that it is imperceptible to the human eye(Fig. 11).

- Security, if the attacker gets the hint of a hidden message in the cover image, even then he must not be able to extract it and use it for his personal gain.

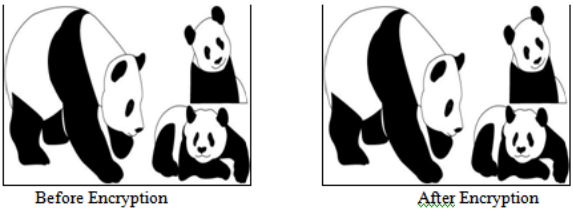

Fig. 11. Comparison between Images before and after encryption

Considering the above criterions our algorithm was found to be quite satisfactory since as relevant from the Fig. 11, the distortion in the image after undergoing the data embedding is negligible and due the features like spreading the data and variable bit embedding, the extraction of data by an unintended receiver is also quite difficult.

5 Future Prospects

In our algorithm we will like to modify certain things which includes using of asymmetric key instead of using symmetric key during pixel selection and we will try to embed secret message bits in image formats other than bitmap as well .Also we will try make our algorithm compatible for Video Steganography[8] and Audio Steganography[5] as well .Also we will try to reduce the distortion in the image when asymmetric keys will be used in our algorithm.

6 Conclusion

Steganography has its place in security. It is not intended to replace cryptography but supplement it. Hiding a message with Steganography method reduces the chance of a message being detected. However, if that message is also encrypted, if discovered, it must also be cracked (yet another layer of protection). In this work we have tried to develop a new algorithm for Steganography and the main objective of the algorithm is not only to hide the data but also to make sure that the data remains hidden and even if by some means an attacker get to know about the presence of data in the cover image even then he must not get the message itself.

References

1. The Wiki (2008), http://www.en.wikipedia.org/wiki/Steganography
2. Chandramouli, R., Memon, N.: Analysis of LSB based image steganography techniques. In: Proc. of ICIP, Thessaloniki, Greece (2001)

3. The Wikipedia, http://www.en.wikipedia.org/wiki/Steganography_tools
4. http://www.ims.nus.edu.sg/Programs/imgsci/files/memon/sing_stego.pdf
5. Johnson, N.F., Duric, Z., Jaljodia, S.: Information Hiding – Steganography and Watermarking – Attacks and counter measures, USA (2001)
6. The Gary Kessler, http://www.garykessler.net/library/crypto.html
7. The Difference between, http://www.differencebetween.com/difference-between-cryptography-and-vs-steganography/
8. The Connect geeks, http://www.connectgeeks.com/?p=589