International Journal of Advance Research in Engineering, Science & Technology(IJAREST),
ISSN(O):2393-9877, ISSN(P): 2394-2444,
Volume 2,Issue 12,December-2015 Impact Factor: 2.125

Data hiding using Hide & Seek Technique

Chirag M. Patel[1], Amit Agrawal [2], Reena M. Patel [3]

[1]Electronics & Communication Dept., Silver Oak college of Engineering And Technology
[2]Electronics & Communication Dept., Silver Oak college of Engineering And Technology
[3]Electronics & Communication Dept., LDRP Institute of Technology & Research

Abstract

Steganography is an art of secret data communications. Its main purpose is to hide the occurrence of communication over a public channel. Steganography tends to hide the very existence of the message or any communication form. Hiding the occurrence of communication can be done by embedding a secret message into an innocent cover medium, such as an image, which no one else than the sender and the recipient can suspect. There could be some important data that need to be protected during transmission. Therefore, how to protect the secret messages during transmission becomes an important research issue. In this paper author has discuss out least significant bit base algorithm and check quality of image. Here also discuss about attacks. In this paper we find some new techniques to overcome the attacks problem.

Keywords- *Steganography, capacity, Delectability (perceptibility), Robustness, attack, Stegnoanalsis*

I. INTRODUCTION

All manuscripts must be in English. The term Steganography refers to the art of covert communications [5]. By implementing steganography, it is possible for Alice to send a secret message to Bob in such a way that no-one else will know that the message exists. Typically, the message is embedded within another object known as a cover work, by tweaking its properties. The resulting output, known as a stegogramme is engineered such that it is a near identical perceptual model of the cover work, but it will also contain the hidden message. It is this stegogramme that is sent between Alice and Bob. If anybody intercepts the communication, they will obtain the stegogramme, but as it is so similar to the cover, it is a difficult task for them to tell that the stegogramme is anything but innocent. It is therefore the duty of steganography to ensure that the adversary regards the stegogramme and thus, the communication as innocuous.[1]

When a steganographic system is developed, it is important to consider what the most appropriate cover Work should be, and also how the stegogramme is to reach its recipient. With the Internet offering so much functionality, there are many different ways to send messages to people without anyone knowing they exist. For example, it is possible that an image stegogramme could be sent to a recipient via email. Alternatively it might be posted on a web forum for all to see, and the recipient could log onto the forum and download the image to read the message. Of course, although everyone can see the stegogramme, they will have no reason to expect that it is anything more than just an image. In terms of development, Steganography is comprised of two algorithms, one for embedding and one for extracting.

The embedding process is concerned with hiding a secret message within a cover Work, and is the most carefully constructed process of the two. A great deal of attention is paid to ensuring that the secret message goes unnoticed if a third party were to intercept the cover Work. The extracting process is traditionally a much simpler process as it is simply an inverse of the embedding process, where the secret message is revealed at the end.

The entire process of steganography for images can be presented graphically as; two inputs are required for the embedding process: Secret message - usually a text file that contains the message you want to transfer & Cover Work: used to construct a stegogramme that contains a secret message.

The next step is to pass the inputs through the Stegosystem Encoder, which will be carefully engineered to embed the message within an exact copy of the cover Work, such that minimum distortion is made; the lower the distortion, the better the chances of undetectability. The stego-system encoder will usually require a key to operate, and this key would also be used at the extraction phase. This is a security measure designed to protect the secret message. Without a key, it would be possible for someone to correctly extract the message if they managed to get hold of the embedding or extracting algorithms. However, by using a key, it is possible to randomize the way the stegosystem encoder operates, and the same key will need to be used when extracting the message so that the stegosystem decoder knows which process to use. This means that if the algorithm falls into enemy hands, this extremely unlikely that they will be able to extract the message successfully.

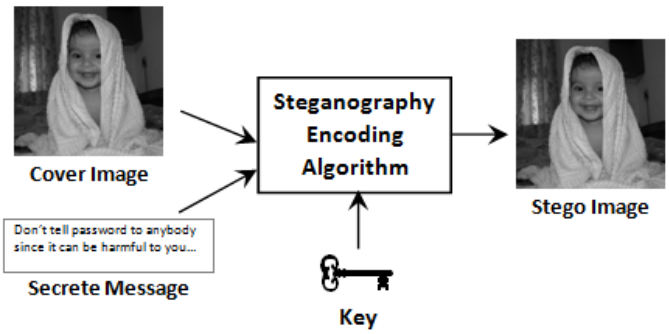

Figure 1: Sender Side of Stego-system

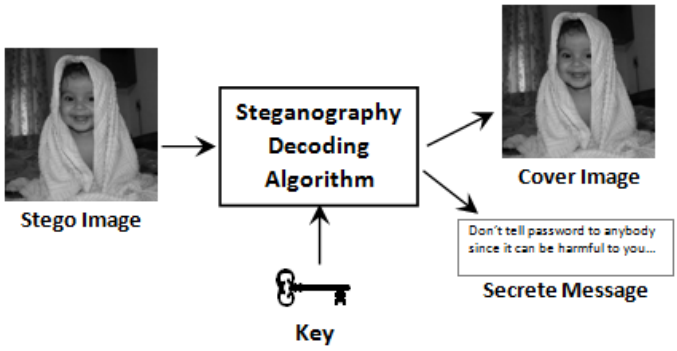

Figure 2: Receiver Side of Stego-system

The resulting output from the stego-system encoder is the stegogramme, which is designed to be as close to the cover Work as possible, except it will contain the secret message. This stegogramme is then sent over some communications channel along with the key that was used to embed the message. Both the stegogramme and the key are then fed into the stego-system decoder where an estimate of the secret message is extracted [2]. Stego-system is shown in fig 1 and 2.

II. PAPER CONCEPT OF STEGANOGRAPGY

There are many embedding techniques available in the field of data hiding, which are used to protect data. Here author has implemented List significant bit technique in Spatial Domain.

LEAST SIGNIFICANT BIT INSERTION

The most common and popular method of modern day steganography is to make use of the LSB of a picture's pixel information. Thus the overall image distortion is kept to a minimum while the message is spaced out over the pixels in the images. This technique works best when the image file is larger then the messages file. Many stego tools make use of least significant bit (LSB). For example, 11111111 is an 8-bit binary number. The rightmost bit is called the LSB because changing it has the least effect on the value of the number.

The idea is that, the LSB of every byte can be replaced with message bit makes a little change to the overall file. The binary data of the secret message is broken up and then inserted into the LSB of each pixel in the image file. The difference between say 11111111 and 11111110 is likely to be undetectable by the human eye. Therefore, the least significant bit can be used. LSB insertion works well with gray-scale images. It is possible to hide data in the least and second least significant bits and the human eye would still not be able to discern it. LSB method is used in Hide & Seek technique.

Substitution Techniques:

- Substitute redundant parts of a cover with a secret message
- Example: Least Significant Bit (LSB) Substitution Choose a subset of cover elements and substitute least significant bit(s) of each element by message bit(s)
- Message may be encrypted or compressed before hiding
- A pseudorandom number generator may be used to spread the secret message over the cover in a random manner
- Easy but vulnerable to corruption due to small changes in carrier

III. HIDE & SEEK TECHNIQUE

The simplest form of image steganography is the method known as Hide & Seek in Sequential approach which replaces the LSBs of pixel values (also referred to as the spatial domain) with the bits from the message bit stream. This algorithm is perform mainly by two ways, first one is sequentially & other one is randomly, author has implemented both the above type for 8 bit & 24 bit.

```
for i = 1, ..., l(m)
  do
     p ← LSB(Ci)
     if p ≠ mi then
        ci ←mi
     end if
  end for
```

Figure 3: Hide & seek encoding process

The encoding process (as shown in Fig 3) shows that the entire algorithm can be implemented by writing just a few lines of code [3]. The algorithm works by taking the first pixel of the image c_i and obtaining its LSB value (as per line 2 of the Algorithm). This is typically achieved by calculating the modulus 2 of the pixel value. This will return 0 if the number is even and 1 if the number is odd, which effectively tells us the LSB value. We then compare this value with the message bit m_i that we are trying to embed. If they are already the same, then we do nothing, but if they are different then we replace c_i with m_i. This process continues whilst there are still values in **m** that need to be encoded.

The decoding phase is even simpler. As the encoder replaced the LSBs of the pixel values in c in sequence, we already know the order that should be used to retrieve the data. Therefore all we need to do is calculate the modulus 2 of all the pixel values in the stegogramme s, and we are able to reconstruct m as m'_i . Figure 4 shows the algorithm of the hide & seek decoding process.

```
for i = 1, ..., l(s) do
     m'i ← LSB (si)
end for
```

Figure 4: Hide & seek decoding process

Note that this time we run the loop for l(s) instead of l(m). This is because the decoding process is completely separate from the encoding process and therefore has no means of knowing the l(m).

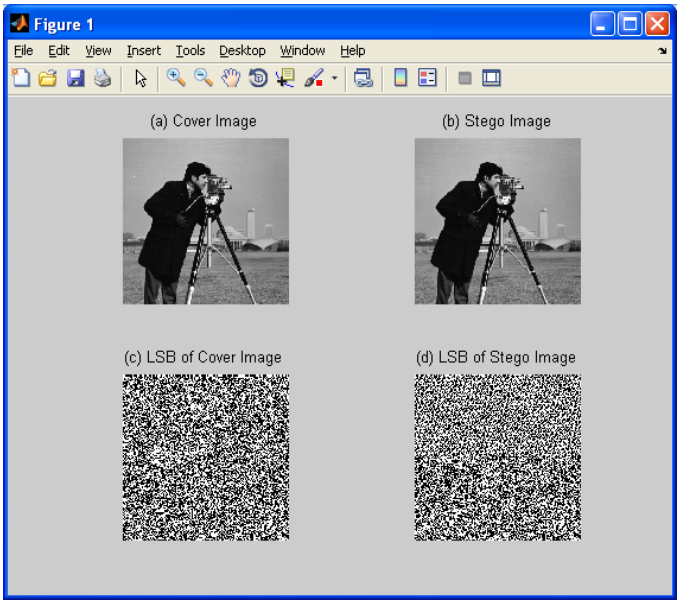

Figure 5: Result of Hide & seek in sequential approach

If a key were used, it would probably reveal this information, but instead we simply retrieve the LSB value of every pixel. When we convert this to ASCII, the message will be readable up to the point that the message was encoded, and will then appear as garbage when we are reading the LSBs of the image data.

Draw back of the Hide & Seek in Sequential approach is that suspect image can be easily detected by visual attack. So the above problem can be overcome using Hide & Seek in Randomizes approach. In Randomize approach data is stored in random location using a PRNG.

generate randomized sequence C using data c & seed k
for i = 1, ..., l(m) do
p ← LSB(Cᵢ)
if p ≠ mᵢ then
cᵢ ←mᵢ
end if
end for

Figure 6: Encoding of randomized Hide & seek algorithm

For the randomized approach the image data c is usually shuffled using a Pseudo Random Number Generator (PRNG). This generator will take the image data c and produce a shuffled version C according to a seed k that is specified by the encoder. There will also be an inverse shuffle which takes C and returns the original order c when the same k is used. The pixel values of the image c are often shuffled before embedding such that the exact same encoding mechanism is used[6]. A Fig 6 shows the pseudo code for the encoding process of the randomized Hide & Seek approach. Note how the bulk of the encoding process remains the same as for the sequential embedding approach. However, we now have line 1 that randomizes the locations of each pixel before embedding the message data. In addition to this, we also have line 8 which returns the pixel locations back to normal when the embedding process has ended. The seed k acts as a key to the algorithm such that the same shuffle sequence can be generated when retrieving the hidden message.

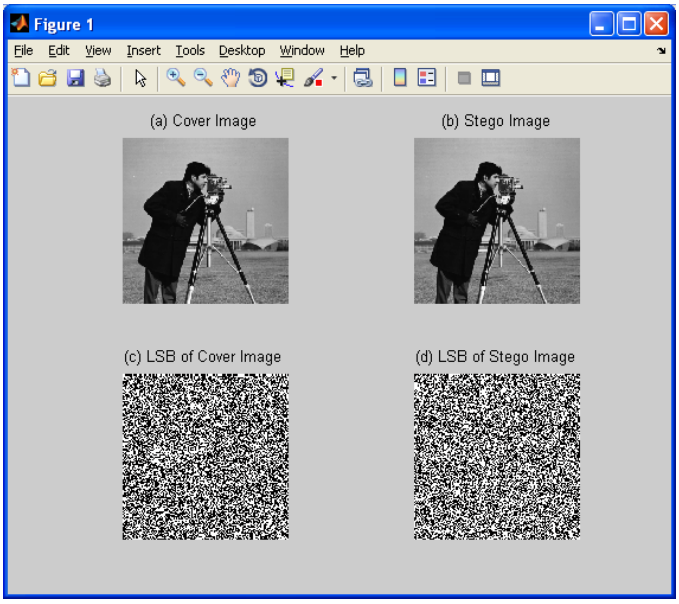

Figure 7: Result of Hide & seek in sequential approach

The output stegogramme s from this embedding approach will contain bits of the hidden message in seemingly random locations of the image.

Perhaps the most important aspect of note is that as we require k to identify the correct regions, the algorithm is much more secure than the sequential approach, as the sequence cannot be derived without it.

IV. RESULT AND CONCLUSION

Image is made from pixel. And every pixel contains 8 bits. Higher most bits are contain maximum information, so if any change is occur in MSB bits it effect the image clarity. But LSB contain less information then MSB bits. So if any change in LSB bits , it not more affect the image[4]. The manipulation of the LSB bit alters the color of the pixel, but this manipulate change can not easily be perceived by the human eye. Lest significant bit insertion using hide & seek algorithm is used to hide maximum data in image. Because each pixel can be used to store data. For the randomized approach the image data is usually shuffled using a Pseudo Random Number Generator (PRNG). This generator will take the image data c and produce a shuffled version C according to a seed k that is specified by the encoder. Hide & Seek in sequential approach has a Only the drawback that it is easily detected by Visual Attacks. When attack is applied if image contain any data then due to visual attach horizontal line is occurred on retrieved image and third person can easily understand the existence of message. That problem can over come using randomize approach. In randomizes approach, data is hide in random manner in image.

ACKNOWLEDGMENT

I would like to express my gratitude to all those who gave me the possibility to complete this paper. I would like to thank Electronic & Communication Department of Silver Oak College and LDRP – ITR, for giving me support. I am bound to my parents for their stimulating support and encouragement which helped me in all the time of writing this paper.

REFERENCES

[1] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press. Simmons, G. J. (1984) 'The prisoners' problem and the subliminal channel', in Advances in Cryptology:Proceedings of Crypto 83 (D. Chaum, ed.), pp. 51–67, Plenum Press

[2] Brainos II, A.C. 'A Study of Steganography and the Art of Hiding Information', East Carolina University.

[3] A. Westfeld. "Detecting Low Embedding Rates", LectureNotes in Computer Science, vol. 2578, pp. 324-339,2003

[4] Andreas Westfeld, Andreas Pfitzmann: Attackson Steganographic Systems, in Andreas Pfitzmann (Ed.): Information Hiding. Third International Workshop, LNCS 1768, Springer-Verlag Berlin Heidelberg 2000. pp. 61–76

[5] "Information Hiding by Steganography and Watermarking", Dr. Mohammed Al-Mualla and Prof. Hussain Al-Ahmad Multimedia Communication and Signal Processing (MCSP), Research Group Etisalat College of Engineering.

[6] "Image Steganography and Steganalysis", Philip Bateman and Dr. Hans Georg Schaathun: lecture note of Department of Computing Faculty of Engineering and Physical Sciences University of Surrey Guildford Surrey United Kingdom, August 2008