Designing an Embedded Algorithm for Data Hiding using Steganographic Technique by File Hybridization

G. Sahoo[1] and R. K. Tiwari[2]

[1] Department of Computer Science &Engineering. , B.I.T., Mesra, Ranchi, Jharkhand, India
[2] Department of Computer Science &Engineering, R.V.S. College of Engg. & Tech., Jamshedpur, Jharkhand, India

Summary
The growing possibilities of modern communication need the special means of security especially on computer network. Data security in the last few years has gained a wider audience. In this paper we have discussed a new steganographic technique based on the file hybridization. In contrast to other methods of steganography where data embedding in image work on the principle of only one image file, the proposed method works on more than one image. The effectiveness of the proposed method is described pictorially and also has been shown that a multi-level of security of data can be achieved.

Key words:
Container file, Supporting file, Stego-key, *Multiple Container file, Hybridization.*

1. Introduction

The main goal of steganography is to communicate message securely in a complete undetectable manner [3, 4, 10]. It has been emerged as a skill of concealing private information inside a carrier that can be considered for all intents and ideas [2, 8]. Digital technology gives us a new way to relate steganographic techniques including hiding information in digital images. It not only goes well beyond simply by embedding a text in an image, but also pertains to other media, including voice, text, binary files and communication channels [1]. The problem of unauthorized copying now-a-days is of great concern especially to the music, film, book and software publishing industries. To overcome such type of problems, some invisible information can be embedded in a digital media in such a way that it could not easily be extracted without any specialized technique [4, 9]. In the other side to it, cryptography is one of effective solution to protect the data from the unauthorized users. It can be mentioned here that although by the use of cryptographic techniques we can convert the data into a cipher text, which is in unreadable format, we can not hide the existence of the same data. Therefore, cryptography in alone is not sufficient to protect the data against the unauthorized access by unauthorized users [4, 5].

It can be mentioned here that steganography, in respect to constructive approach, generally provides vast potential security and legal data hiding. As far as corporate sectors are concerned, if they can not hide their important information, this may lead to damage extremely the company's profitability as well as its sustainability. Again, as a destructive approach, even if steganography is not a threat in general, it may have some bad impacts on the areas where one wants to deal with some kind of unwanted illegal matters.

In the above respect, a number of techniques have been developed [1,7] using features like

- Substitution
- Masking & Filtering
- Transform Technique

The method of substitution generally does not increase the size of the file. Depending on the size of the hidden message, it can eventually cause a noticeable change from the unmodified version of the image [4,6]. Least Significant Bit (LSB) insertion technique is an approach for embedding information in a cover image. In this case, every least significant bit of some or all of the bytes inside an image is changed to a bit of the secret message. When using a 24-bit image, one bit of each of the primary color components can be used for the above purpose. It can be seen that on an average only a half of the bits in an image will need to be modified to hide a secret message using the maximum cover size. Since there are 256 possible intensities of each primary color, changing the LSB of a pixel result in only small changes in the intensities of the color. As human eyes cannot perceive these changes, one can say that the message is successfully hidden.

In this context, the masking and filtering techniques, starts with the analysis of the image. Next, we find the significant areas, where the hidden message will be more integrated to cover the image and lastly we embed the data in that particular area. There fore, while in the case of LSB techniques all the least significant bits are changed, in masking and filtering techniques the changes can take place only in selected areas.

In addition to the above techniques for message hiding, transform techniques has also played some important role in embedding the message by modulating coefficients in a

Manuscript received January 5, 2008
Manuscript revised January 20, 2008

transform domain. As an example, Discrete Cosine Transform works by using quantization on the least important parts of the image in respect to the human visual capabilities.

In conclusion, we can say that all of the above methods used for embedding data in an image work on the principle of one image file. In this work, we consider that a single image file may be divided into two or more sub image files and based on our requirements we can embed the required data into a particular chosen sub image file which is, of course, a part of mother image file. This concept helps us in designing a methodology for both hiding and extracting information and is discussed in the following section. In section 3, we have described the case of multiple container file followed by comparison and conclusion in the subsequent sections.

2. Hybrid File Creation

The concept of hybridization may be used in the field of steganography, where more than one file is to be merged and a new hybrid file may consequently be generated. This hybrid file basically consists of two files, namely

- Container file
- Supporting file

Container File: As the name suggests this is a file where we can store the secret data. The basic property of this container file is such that even if we change the intensity of any pixel it should look like the original image. Appropriate candidate for this purpose are cartoon images, geographical images, background images of any picture or images in any chemical reaction and like others. For example, in the case of twinkling star or any other geographical image we can see that by an appropriate change in colour of the object lying in the image can give the same impression of the original image. In order to store the secret data in such a file, the size of the container file should be proportional to the size of the secret data.

Supporting File: To make the image common, we need a supporting image file so that the new hybrid file looks like the original one. The selection of supporting file will depend on the feature of the container file to ensure the above characteristics.

There will be two options in this process; either we can put the container file into the supporting file or vice versa.

2.1 Container File Creation

There are three different aspects in information hiding system: capacity, security, and robustness. Capacity refers to the amount of information that can be hidden in the cover medium; security refers to an eavesdropper's inability to detect hidden information whereas the robustness is to refer to the amount of modification that can be made such that the stego medium can withstand before an adversary can destroy the hidden information. The basic steps for creating the container file is shown in Fig.1, where message is the secret data, the sender wishes to keep it confidential, is the input to the encryption algorithm; stego-key is nothing but used for the required password; the encryption algorithm performs various substitutions and transformations on the plain text.

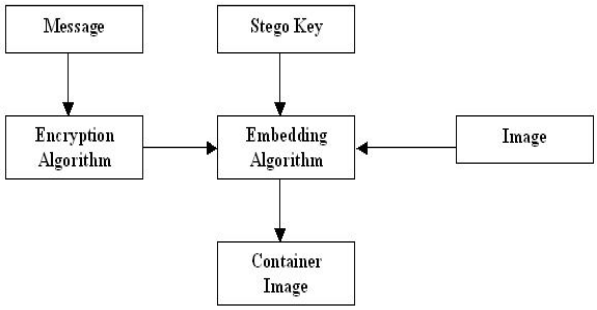


Fig. 1 Basic Model for Container file Creation

2.2 Proposed Embedding Algorithm

In contrast to the LSB scheme being used for hiding the data, our main purpose here is to consider the entire byte representing any particular pixel for storing the information. Consequently, by the process of replacing the entire byte for embedding information in the container image, only one pixel can be used to store three characters, whereas by LSB technique one need a minimum of nine pixels to store these three characters as three pixels are required to store a single character. Further, if we replace all pixel values of an image then the entire image generally changes and may look completely like another image of suspicious. However, by the use of the concept of hybridization both the supporting file and the container file together give the impression of a normal unsuspicious image file.

Again when the volume of data to be sent is comparatively less than the container image file, instead of choosing all pixels for replacement a selected number of pixels can be considered for our purpose. In this case, the selection criteria of choosing the pixels have been done here with the help of mathematical expression as discussed below.

If we consider the size of an image as M x N, and Z be the amount of secret data that to be sent through this image then, the relation between the size of data and the place in the container file may be defined as follows.
For the former case, certainly we have

$$Z < 3 \times M \times N \quad \ldots (1)$$

For the later case, to distribute data identically in image we consider the straight line formula defined as

$$y = ax + b \quad \ldots (2)$$

where the values of a and b play an important role in correspondence to the size of the data to be embedded.
Consider P as the maximum space required for storing the secret data of size is Z, we explain the following cases here just for illustration purpose.
Case I: **When** Z = P.
In this case, the constants a and b take the value 1 and 0 and gives the equation (2) as

$$y = x \quad \ldots (3)$$

It means that each byte of secret data will be replaced by only one byte of the image file.

Case II: **When** Z < P.
When the size of secret data is less than the image space, the value of a and b will be chosen in such a way that the all secret data can be accommodated in the considered image file. Taking a = b = 2, we have

$$y = 2x + 2 \quad \ldots (4)$$

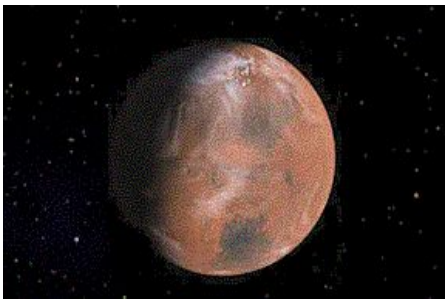
Here, if the value of x=1 then y will be 4, indicating that the first byte of secret data will be stored in the first byte place of second pixel. i.e. (8 – bit Red position). Similarly, x=2 will give y=6 so the second secret byte data will be stored in last byte of second pixel (i.e. 8-bit Blue position) and so on. Hence, by a suitable choice of a and b all bytes of the secret data can be mapped entirely into the container image.

Case III: **When** Z > P.
If the secret data is more than the image space, then it is not possible to store all the secret data into the image file, which is trivial.

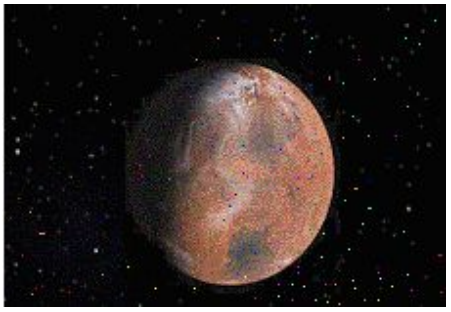
The corresponding pseudo code for generating the container file based on the above analysis is given below and can be visualized from Fig. 2. The data that is inserted into the container file is the sentence" The main goal of steganography is to communicate in unseen mode".

/* Code for generating Container File */

```
{
    Array source_image_buffer: = LoadImage ( "sourec_image" , & totalLenght);
    String  secret_txt_buffer := LoadText ( "Secret_Data" );
    String secret_ascii_buffer;
    Int a,b,x,y;

    Input a,b,x;

    for index = 0 to Length ( secret_txt_buffer)
      secret_ascii_buffer [index]= ASCII (secret_txt_buffer [index]);
    end for;

   for i = 0 to Length (sourceimage) step 3
         y = a * x + b;
              If ( y = x ) then
               {
        GetRGB (&Red, &Green, &Blue, sourceimage[i]);
        Encode (secret_ascii_buffer[i],Red);
        Encode (secret_ascii_buffer[i+1],Blue);
        Encode (secret_ascii_buffer[i+2],Green);
        SetRGB(Red, sourceimage[i]);
        SetRGB(Green, sourceimage[i+1]);
        SetRGB(Blue, sourceimage[i+2]);
                x=x+3;
              }   end if
     end for
  SaveImage( "New_Image" )
}
```



a) Container file before inserting the data



b) Container file after inserting the data

Fig.2 Creation of Container file

After creation of the container file the next step is to create the hybrid file.

2.3 Hybrid File creation and Data Extraction

Arithmetic or Logical operations can be performed on pixel-by-pixel basis between two or more images. For an example, subtraction of two images result in a new image whose pixel at co-ordinate (x, y) is the difference between the pixels as the same locations in the two images being subtracted. Depending on the hardware or software being used, the actual mechanism of implementing arithmetic/ logic operation can be done sequentially. In this method, first a supporting image is selected and then based on our requirements (i.e. the size of message) we select the appropriate container file. If we consider the size of the supporting image as, M_1 x N_1, then we can place the container file inside the supporting file in a region defined by A(x, y) and B(x + r, y + s) for a suitable value of r and s, where $0 \le r \le M_1$ and $0 \le s \le N_1$.

Since we are hybridizing the file, pixel of the supporting image file will be replaced by the corresponding pixel of the container file. If we denote S and C as the set of pixels for supporting and container files respectively, then we can write

$S = \{ x_{ij} \}$, $0 \le i \le M_1$ and $0 \le j \le N_1$,

$C = \{ y_{lk} \}$, $0 \le l \le M$ and $0 \le k \le N$.

The transformation can take place for some value of r and s as (Fig.3)

$y_{r+i, s+j} \leftarrow x_{i,j}$
where $0 \le r \le M_1 - M$ and $0 \le s \le N_1 - N$.

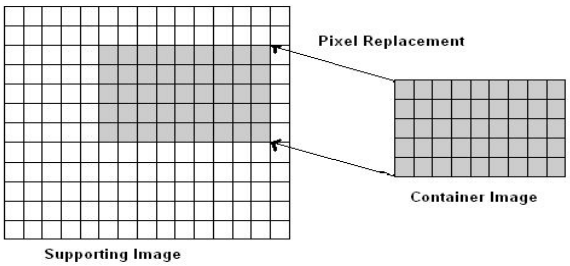
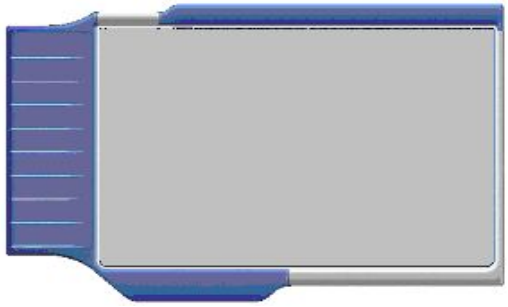


Fig. 3 Merging Container file to supporting file.

After replacing all pixel values of supporting image by the container image we get the resultant hybrid or mixed image (Fig. 4).



a) Supporting File

b) Hybrid Image File after merge of a) and Fig. 2 (b)

Fig. 4 Formation of Hybrid Image File

2.4 Extraction algorithm

The embedded data can be retrieved back by using the following extraction algorithm.

/* Data Extraction */

```
{
  Array source_image_buffer: = LoadImage ( "hybrid_image" , & totalLenght);
  String  secret_txt_buffer
  String secret_ascii_buffer;
  Int a,b,x,y;

 Input a,b,x;

 while eof (hybridimage)
   {
     y = a * x + b;
  If ( y = x ) then
    {
       GetRGB (&Red, &Green, &Blue, sourceimage[x]);
       Decode (secret_ascii_buffer[x],Red);
       Decode (secret_ascii_buffer[x+1],Blue);
       Decode (secret_ascii_buffer[x+2],Green);
       x = x + 3;
    }
  endif
  }

  for index = 0 to Length ( secret_txt_buffer);
    secret_ascii_buffer [index]= CHR (secret_txt_buffer [index]);
  endfor

}
```

3. Multiple Container File

From above, it is seen that the size of the information to be under lock and key relatively depends on the size of the container image. That is, the messages are of smaller size than that of the container file. This concept can easily be extended to answer the following queries. What happens if size of the message is larger than the size of the container image file or how can a number of users (receiver) be entertained with different but similar kind of information? Suppose we have a number of suitable container files where, each individual file is not sufficient to hold the amount of data to be sent. In this case, one has no other choice than choosing more than one such container files. If so, these files can be joined together to form a large container file. Further, if we want to send different sets of data items to either a single user or a number of users we can also combine a required number of appropriate container files to have an unsuspicious container file of the desired size. Such multi-container file demands the use of different stego-key for each component of the above file. It gives an extra layer of protection. One such hybrid multi-container file is shown in Fig.5.

Fig. 5. Hybrid Image File with Multi Container File

4. Comparison

The performance of the proposed technique has been highlighted here with the concept being used in the case of Least Significant Bits (LSB) methods. As the name suggests, the information is stored here by changing the least significant bits of a required number of bytes of the image. Although, this simplicity of replacement gives the advantage of the LSB techniques, in the other hand the message can easily be destructed by the attacker by interchanging and/or placing either zero or one at the respective least significant positions. Even if a maximum of four least significant bits can be replaced by these techniques, which is evident from the literature, a fruitful result may not occur due to the complexity of the nature of the replacement. In contrast, since the proposed method works on the principle of replacing an entire byte for a particular character of the message to be sent, a similar change by an attacker may not obstruct to retrieve the original message by a suitable adjustment. For example, if the least significant bit in the code of a character is changed, then definitely the changed character is either the preceding character or the following character of it in the respective alphabet. We hope, the corrected preceding characters as a whole belonging to any valid word can suitably match the next character by changing it, if any, by the help of a dictionary. Hence, in this case retrieving the original message from the changed message, if found, is much easier than the previous cases.

5. Conclusion

The suitability of steganography as a tool to conceal highly sensitive information has been discussed by using a new methodology sharing the concept of hybridization and a multilevel of security of data is achieved. This suggests that an image containing encrypted data can be transmitted to any body any where across the world in a complete secured form. Downloading such image and using it for many a times will not permit any unauthorized person to share the hidden information. Industries like music, film, publishing and organization like ministry and military will definitely be highly benefited by the use of such techniques. We can conclude here by saying that combination of both steganography and cryptography can provide us a double layer of protection.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable suggestions.

References

[1] N.F. Johnson and S. Jajodia, " Exploring Steganography: Seeing the Unseen ", Computer, vol. 31, No. 2, Feb 1998, pp. 26-34.
[2] Donovan Artz " Digital Steganography: Hiding Data within Data", IEEE Internet computing, pp.75-80 May –June 2001.
[3] Niels Provos and Peter Honeyman, "Hide and Seek : An Introduction to Steganography", IEEE 1540-7993, June 2003
[4] M.M Amin, M. Salleh, S. Ibrahim, M.R. Katmin, and M.Z. I. Shamsuddin " Information hiding using Steganography" IEEE 0-7803-7773-March 7, 2003.
[5] Kefa Rabah, "Steganography – The Art of Hiding Data", Information Technology Journal 3 (3): 2004, ISSN 1682-6027, 2004 Asian Network for Scientific Information..

[6] Bret Dunbar, "A Detailed Look at Steganography techniques and their use in an Open Systems Environment ", January 18, 2002 SANS Institute.
[7] F.A.P Peticolas, R.J. Anderson and M.G.Kuhn, "Information Hiding –A Survey", and Proceedings of IEEE, pp. 1062-1078, July 1999.
[8] Venkatraman. S, Ajith Abraham and Marcin Paprzycki " Significance of Steganography on Data Security ", IEEE 0-7695-2108-8, 2004.
[9] W. Bender, D. Gruhul, N. Morimoto, and A. Lu. "Technique for data hiding", IBM Systems Journal, 35(3 & 4), 1996.
[10] R. van Schyndel, A. Tirkel, and C. Osborne. "A digital water Mark.", Proceeding of the IEEE International Conference on Image Processing, 2:86-90, 1994.

G. Sahoo received his P.G. degree from Utkal University in the year 1980 and Ph.D degree in the area of Computational Mathematics from Indian Institute of Technology, Kharagpur in the year 1987. He is associated with Birla Institute of Technology, Mesra, Ranchi, India since 1988. He is currently working as a professor and heading the Department of Computer Science and Engineering. His research interest includes theoretical computer science, parallel and distributed computing, data security, image processing and pattern recognition.

Rajesh Kumar Tiwari received his M.C.A degree from Nagpur University in the year 2002. Currently, he is Asst. Professor in R. V. S. College of Engg. and Technology, Jamshedpur, India. His research is focused on data security.