International Journal of Information Technology and Knowledge Management
July-December 2008, Volume 1, No. 2, pp. 209-217

DESIGNING SOME IMPERCEPTIBLE DATA HIDING METHODOLOGIES USING STEGANOGRAPHIC TECHNIQUES

R. K. Tiwari & G. Sahoo

Abstract

Data hiding is the process of embedding the secret data in cover image by making small modification to its pixels. The different methods of steganography are mostly applied on image files to embed the secret data. All these methods replace only some parts of the chosen pixels and due to this only small data can be embedded in cover image otherwise; a large embedding data changes the cover image originality. So, one has to compromise either in size of the secret data or in the visual quality of cover image. This paper presents a new methodology for embedding a large embedding data with very small changes in respectively small cover image. Based on this method, we have constructed setgo image creator and secret data viewer in Microsoft platform.

Keywords: Stego-key, User Classified Multi-valued Code, Reduced ASCII code, Direct Replacement.

1. Introduction

The growing use of internet need to store, send or receive personal information. The common approach is to transfer the data in different form so that the resultant data can be understood only by those who can get it back into its original form. This technique is known as encryption. A major draw back of this method is that the existence of data is not hidden. If someone gives enough time then he may get the unreadable encrypted data in its original form. The concept of steganography has considerably been given much attention in recent years in order to overcome such type of problems. The primary aim of steganography is to hide data in a cover media so that other will not notice it [9, 10]. The characteristics of a cover media depends on the volume of data that can be hidden, the perceptibility of the message and its robustness [4, 5, 6, 8, 10].

This area of research work has a great impact on the fields like publishing and broadcasting where, hiding information needs much attention. Unauthorized copying also becomes a hot issue in the area like music, film, book and software. To overcome such type of problem some invisible information can be embedded in the digital media in such a way that no one easily extract the required information or message [1, 2, 4]. Further, software industries also have taken advantage of another form of steganography,

called watermarking. It is used to establish ownership, identification, and provenance [3, 7].

2. Related Works

The most suitable cover media for using Steganograpic techniques is an image. Several methods have already been discussed in the literature. The main reason behind it is the large redundant and the possibility of hiding information in the image without attracting attention to human visual system. Feature like substitution, masking and filtering and transform techniques have been considered in the development of such techniques [1, 7].

Note that 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]. In this regard, the Least Significant Bit (LSB) insertion technique is an approach for embedding information in a cover image where, every least significant bit of some or all of the bytes inside an image is changed to a bit of the secret message. If we use a 24-bit image, one bit of each of the primary color components can be used for the above purpose. This can imply 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 the context of masking and filtering techniques, starting with the analysis of the image and then, we find the significant areas where, the hidden message will be more integrated to cover the image and finally we embed the data in that particular area. In contrast to LSB technique where all least significant bits are changed, we can just say here that in masking and filtering techniques changes 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 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. Raja [11] concluded that three secret bits can be embedded in one pixel. Anderson [12] observed that the secret message in JPEG images can be embedded in multiple locations. Amin [4] has also given a secure information hiding system which can embed 60 KB message in 800 x 600 pixels image. Embedding secret data in an image works on the principle of replacement of entire or some parts of the chosen pixels. Due to this replacement policy however, we are not able to embed the high capacity secret data. In the other side, if we replace all pixel values of cover image then the stego image may look like suspicious image. In this paper, we have proposed a new technique in section 3 where, we have discarded the demerits of both systems and

described a new methodology for high capacity data hiding with negligible visual changes for human visual system. For useful applications of this method, an attempt has also been made here to construct stego image creator and secret data viewer which have been discussed in section 4; section 5 gives the conclusion of this paper.

3. PROPOSED METHOD

The basic model of stego image creation is shown in fig. 1. The six interrelated steps creates the stego image are explained below.

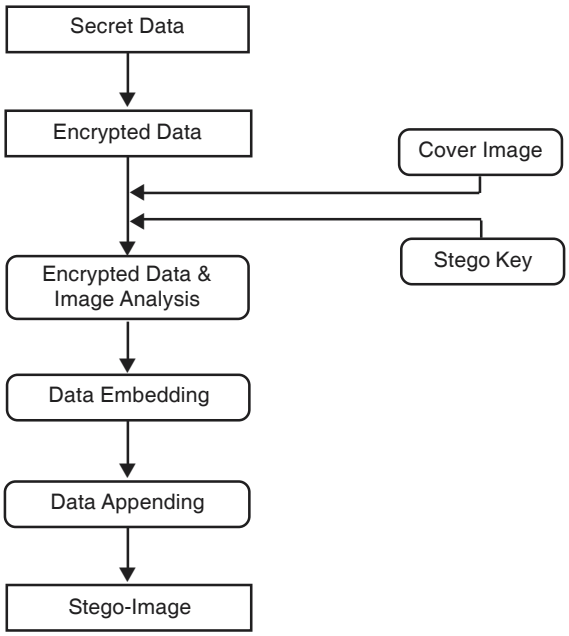

Figure 1: Basic Model for Stego-Image

First Step: Message is the secret data, which the sender wishes to keep confidential. This is the original message or data that is fed into the algorithm as input. This is worth to explain here that we can take a huge data as input.

Second Step: The second foot step begins with the encryption of the secret data, which gives an extra layer of protection.

Third Step: The cover image is a file where we can store the secret encrypted data. The selection of the cover image is user dependent. It should be noted that, we can select a small cover image file for a huge secret data. The stego key is known as password, which is a protection fence for this methodology.

Fourth Step: This phase is very vital and we start here with the cover image and secret data analysis. The size of the secret data is checked and based on that we move to different steps. Since our main aim is to conceal large amount of data with very negligible visual changes of cover image. Analyzing and finding the different portion of cover image where we can directly store the data. This direct replacement of the pixel value to the corresponding secret data character gives a freedom to store three character in one pixel where as, the LSB methods require nine pixel to store same three characters. The portion selection can be either manually or through selection technique. The manual selection of image portion depends on the user. Fig. 2 gives the manual selection and data embedding feature of cover image. The marked area can be taken as good place for direct storing of the secret data. The strong reason behind the embedding is non suspicious stego image. The alternative selection process starts with the formation of histogram. The image histogram is a valuable tool used to view the intensity profile of a cover image. The histogram provides information about the contrast and overall intensity distribution of a cover image. Here we are plotting the most frequently and character matching pixel intensities (Fig. 3). The intensities with nearly matching secret data are directly replaced. By this way a good amount of data can be stored with very less changes. Further, by using the most common LSB approach for data embedding we can boost our embedding capacity.

(a) **(b)**

Figure 2: (a) Cover Image (b) Manual Selection

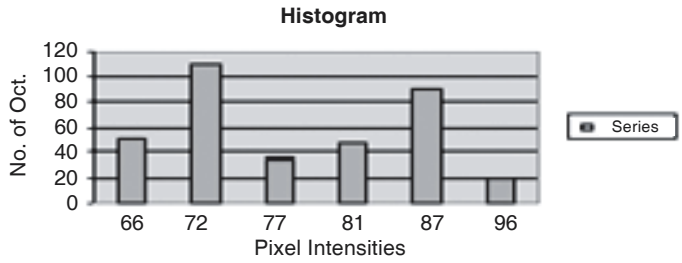

Figure 3: Histogram of Cover Image (Fig. 2)

/* Pseudo Code for Step-IV */

Step IV {Integer Size; Print "Select Option 1 for Manual data place & 2 for Automatic"; Switch (I) Case 1:Call Manual; Break; Case 2: Call Automatic; Break; End Switch; Return Size; }

/ * Manual Place finding and Data Embedding */

Manual {1DArray Cover_iamge = Load ("Cover image", &length); 1D Array Stego_Image [ ]; 1D Array Secret_data [ ]; 1D Array ASCII_data []; Integer L, I, J, K; INPUT Secret_data;

/* Input the Data Store Position */

INPUT L; For I= 0 to Length (Secret_data [ ]) ASCII_data [I] = Secret_data [I]; End For;

If Length (Secret_data [ ]) > Length (Cover_image [ ]) Then Print "Secret Data size is more, Select other method"

Else Pixel_Place=L; J=1; For I =L to Length (Secret_data [ ])

GetColorValue (&K, Cover_image [Pixel_Place]); K= Secret_data [J];

SetColorValue (Stego_image [Pixel_Place], &K); J=J+1; End For; End if;

SaveImage (Stego_image []); RenameImage (Stego_image [], Cover_Image []);

Remove Image (Stego_image [ ]);}

Automatic {1DArray Cover_iamge = Load ("Cover image", &length); 1DArray Stego_Image [] 1D Array Secret_data [ ]; 1D Array ASCII_data [ ]; Integer L, I, J, K, Key_asc; INPUT Secret_data; For i= 0 to Length (Secret_data [ ]) ASCII_data [I] = Secret_data [I]; End For;

/* Place finding and data storing */

For J= 65 to 95 Key_asc=J; For i = 1 to Length (Cover_iamge [ ])

GetColorValue (&K, Cover_image [Pixel_Place]); If Asc (K) = Key_asc Then

Count =Count +1; Else If Count > 50 Then For M=1 TO Count K= Secret_data [J];

SetColorValue (Stego_image [M], &K); End for; End If; Count= 0; End If; End For;

End For; SaveImage (Stego_image []); RenameImage (Stego_image [ ], Cover_Image [ ]);

Remove Image (Stego_image [ ]);}

Fifth Step: Digital images are commonly of two types i) 8 bit images and ii) 24 bit images. These images can logically be divided into two parts name as first part and second part. The first part consist of all preliminary information like file name, file type, compression type etc. and the second portion keeps the information of pixel. If without changing the value of first part we append the secret data at the end of the second part then we may able to store a large amount of data. This leads to get advantages: first there will no any visual changes in cover image and second, a small cover image file may keep a large amount of secret data (see Fig. 4 & 5). Further, the appending of data can be done by using some mathematical formula by which may give more protection of secret data can be achieved. The selections of high intensities images are not often use in the field of steganography, but since we do not change any thing in the preliminary or secondary part so, this step gives the freedom to choose any type of images. The corresponding pseudo code for this step is given below.

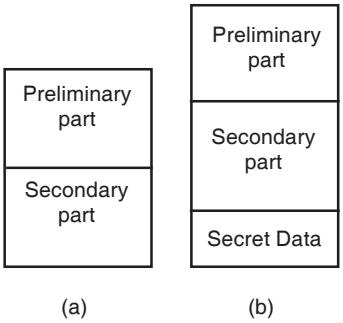

Figure 4: a) Cover Image b) Stego Image

Figure 5: a) Cover Image b) Stego Image with Secret Data

/* Pseudo Code for Step-V */

Step V {1DArrayCover_iamge= Load ("Cover image",& length), Stego_Image[]; 1DArray Secret_data[]; 1D Array ASCII_data [ ]; Integer I, J, M; INPUT Secret_data;

For I= 0 to Length (Secret_data [ ]) ASCII_data [I] = Secret_data [i]; End For;

For I =0 to Length (Cover_image [ ]) GetColorValue (&K, Cover_image Pixel_Place])

SetColorValue(Stego_image[Pixel_Place],&K); End For;

For M =I to Length (Secret_data [ ]) J= ASCII_data [M]; SetColorValue (Stego_image [M], &J); End For; SaveImage (Stego_image []) RenameImage (Stego_image [ ], Cover_Image[ ]);

Remove Image (Stego_image [ ]) ;}

Sixth Step: This last step of the proposed methodology gives the stego image as the out put. In view of the fact that the changes made are very minor so the visual appearances of both cover image and stego image are appeared to be same. This clearly strengthens the concept that has been considered in this work. The three different aspects consider in any 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. In this proposed system we have taken care of all the aspect of information hiding system.

/* Pseudo Code for New System */

Data_Hiding_Method {1DArray Cover_iamge = Load ("Cover image", & length);

1D Array Stego_Image [ ]; 1D Array Secret_data [ ]; 1D Array ASCII_data [] Integer I,

Size = Step IV ( ); If Size= Length (Secret_data []) Then Call Step V;

End if; SaveImage (stego_image []) ;}

The mean square error between the cover image and stego-image can be used as one of the measure for the relative perceptibility of the embedded secret data. The mean square error and peak signal to noise ratio are used here as the measuring parameters for the finding the amount of imperceptibility. The mean square error, MSE, can be defined as

$$MSE = 1/MN\left[\sum_{i=1}^{M}\sum_{J=1}^{N}\left(F_{i,j} - G_{i,j}\right)^2\right]$$

where, M and N are the rows and columns respectively of the cover image. Let $F_{i,j}$ be the pixel value of the stego image. The peak signal noise ratio of the cover image can then be defined as

$$PSNR = 10\ \mathrm{Log}_{10}(L^2/MSE)$$

where, L is peak signal of the cover image and the value of L is 255 for 8 bit cover image. Here, due to the multi value code the changes are very small. The *PSNR* threshold for stego-image in Fig. 6 is 33dB only. The embedded data can be retrieved back by using the extraction algorithm. Generally the extraction is the reverse process of embedding the data in to the image. The extraction algorithm reads the embedded data from the stego-image and with the aid of a grammar and dictionary the secret data is retrieved and fabricated (Fig. 6 & 7).

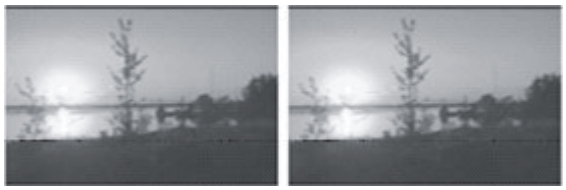

Figure 6: a) Cover Image b) Stego Image

Secret Date	Retrieved Data
The word steganography is a Greek word that means 'writing in hiding'. The main purpose of steganography is to hide data in a cover media so that other will not notice	The word steganography is a Greek word that means 'writing in hiding' The main purpose of steganography is to hide data in a cover media so that other will not notice

Figure 7: Secret Data & Retrieved Data

4. APPLICATION

Based on the above methodology we have designed Stego Image Creator (SIC) and Secret Data viewer (SIV) in Microsoft platform, and it may be implemented in other programming platform like JAVA & C. In the Initial step we are finding the space using the step 4, and if still the secret data is left we move to the fifth steps. Finally we get the stego image. The secret data viewer gives the secret data when the password is correct. Fig.8.

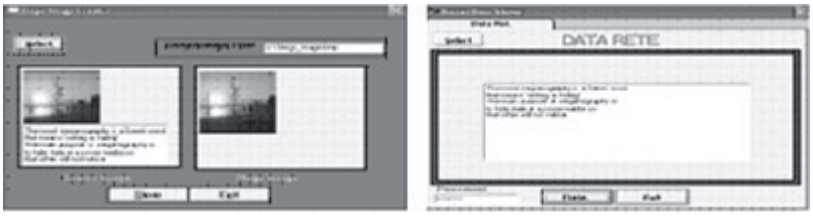

Figure 8: a) Stego Image b) Secret Data Viewer

5. CONCLUSION

The suitability of steganography as a tool to conceal highly sensitive information has been discussed by using a new methodology. This suggests that a small cover image can store the large amount of data very efficiently with negligible visual changes. The cover 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.

6. REFERENCES

[1] G. Sahoo and R. K. Tiwari "Designing an Embedded Algorithm for Data Hiding using Steganographic Technique by File Hybridization", *IJCSNS* **8** (1) January 2008.

[2] Donovan Artz "Digital Steganography: Hiding Data within Data", *IEEE Internet Computing,* May–June 2001, 75–80.

[3] Mitchell D. Swanson, in Zhu and Ahmed H. Tewfik "Transparent Robust Image Watermarking" *IEEE* 0-7803-3258-x/96 (1996)

[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] Lisa M. Marvel and Charles T. Retter, "A Methodology for Data Hiding using Images", *IEEE* 0-7803-4506-1/98,

[6] Bret Dunbar, "A Detailed Look at Steganography Techniques and their use in an Open Systems Environment", January 18, 2002 SANS Institute.

[7] C. Cachin, "An Information–Theoretic Model for Steganography", in *Proceeding 2nd Information Hiding Workshop*, **1525**, (1998), 303–318.

[8] F. A. P Peticolas, R. J. Anderson and M. G. Kuhn, "Information Hiding –A Survey", in *Proceeding of IEEE*, July 1999, 1062–1078.

[9] Venkatraman. S, Ajith Abraham and Marcin Paprzycki, "Significance of Steganography on Data Security", *IEEE* 0-7695-2108-8, 2004.

[10] N.F. Johnson and S. Jajodia, "Exploring Steganography: Seeing the Unseen", *Computer*, **31**, (2).

[11] Ross J. Anderson and Fabien A. P. Petitcoals "On the Limits of Steganography", *IEEE Journal on Selected Areas in Communications*, **16**, (4), May 1998.

[12] K. B. Raja, C. R. Chowdary, Venugopal K. R., L.M. Patnaik "A Secure Image Steganography using LSB, DCT, and Compression Techniques on Raw Images", *IEEE* 0-7803-9588-3/05.

[13] Gadadhar Sahoo and Rajesh Kumar Tiwari "Some New Methodologies for Image Hiding using Steganographic Techniques", *IJIAP* **1**, (1), April 2008. 25–31.

R. K. Tiwari
Department of Computer Science & Engg.
R.V.S. College of Engg. & Tech.
Jamshedpur
Jharkhand, India
E-mail: *rajeshkrtiwari@yahoo.com*

G. Sahoo
Department of Computer Science & Engg.
B.I.T Mesra
Ranchi, Jharkhand, India
E-mail: *gsahoo@bitmesra.ac.in*