International Journal of Engineering Sciences Paradigms and Researches, Vol. 03, Issue 01, January 2013
ISSN (Online): 2319-6564
www.ijesonline.com

Two Layer Provision for Secure Data Transmission

Dr. Grima Dhingra[1], Ms. Sumity[2]

[1]A. P., Deptt. of Physics, MDU, Rohtak
grimadhingra@gmail.com

[2]Deptt. of Physics, MDU, Rohtak

Abstract

A new security technique for data transmission called dual layered approach is proposed in this paper. It uses the combination of two data security tradeoffs. In first layer data is encoded by application of key encryption and then next layer performs the hiding of encoded data in an image. It uses the same principal of LSB replacement with a new parameter of selection of pixels on basis of their intensity. Pixels of image used for embedding and replacement are selected on basis of the color value of their surrounding pixels and thus more intense and different pixel is selected each time foe hiding data each time in this approach.

Keywords: *Cryptography, Intensity, Least Significant Bit, Steganography.*

1. Introduction

Since the rise in need of communication over internet and latest advancement in digital technologies a large amount of data is being exchanged over internet. Prevention of data from unauthorized access is the main problem. Thus a huge amount of cryptographic as well as steganographic algorithms has been developed for encoding and hiding the message. Although it can be argued that there is no single encryption algorithm satisfies the need of security completely. However several methods have been proposed for the image steganography also but each is associated with certain limitations .In random pixel manipulation technique to perform steganography pixels are selected in random fashion on basis of key decided thus leading to the key management overhead. Whereas in the stego color cycle technique the detection of data is easy and symmetric. Thus no single steganography technique alone is not compatible for fully secure data.

In this paper we propose a new technique that combines the two security approaches cryptography as well as steganography for hiding data in RGB images. However these two differ with each other. Steganography is the art and science of communicating in a way which hides the existence of the communication. In contrast to cryptography where the enemy is allowed to detect intercept and modify messages without being able to violate certain security premises guaranteed by a cryptosystem.

The goal of steganography is to hide messages inside the other harmless messages in the way that does not allow any enemy to even detect that there is a second message present. In this approach data to be transmitted is firstly encoded using key encryption scheme and then compression of data is made in order to reduce the size of data. In the next layer the data is embedded in an image. Embedding of data in image contains the process of replacement of LSB of pixels of image with that of encoded data bits. Pixels of container image are selected for replacement on the basis of the intensity value of the surrounding pixels. While performing this certain specific assumptions are made such as degradation of cover data must be kept to minimum and hidden data must be made as immune as possible from the manipulation of cover data. This technique can be used for fix number of bits per pixel but it can be extended further to more capacity for cover media. The rest of paper is organized as follows. Section 2 deals with the certain issues necessary for reviewing image steagnography as well as image steganography trade offs. Section 3 discusses about the methodology and algorithmic approaches. Section 4 explains experimental results and Section 5 is conclusion.

2. Concepts in Image Steganography

In essence, image steagnography is about exploiting the limited powers of the human visual system (HVS). Within reason any plain text, cipher text, other images, or anything that can be embedded in a bit stream can be hidden in an image. To a computer, an image is an array of numbers that represent light intensities at various points, or pixels. These pixels make up the image's raster data. An image size of 640 by 480 pixels, utilizing 256 colors (3 bytes per pixel) is fairly common. Such an image would contain around 300 kilobits of data.

Bitmap images consist of matrices of numbered points with two dimensions for grayscale and three dimensions for RGB color images. The grayscale images, also called intensity images, contain numbered values at these points called pixels, between 0 for black and 255 for white, which can be represented as 8-bit binary strings (28= 256). The numbers between represent gradient gray values between black and white. The RGB images are actually three two-dimension image layers, a red, a green, and a blue layer that are combine to produce full color image.

The product of these three layers can produce over 16 million different colors and is called 24-bit color because 2563= (28)3 = 224 = 16777216 in which three 8-bit binary strings represent pixel colors. Digital images are typically stored in either 24-bit or 8-bit per pixel files. 24-bit images are sometimes known as true color images. Obviously, a 24-bit image provides more space for hiding information; however, 24-bit images are generally large and not that common. A 24-bit image 1024 pixels wide by 768 pixels high would have a size in excess of 2 megabytes. As such large files would attract attention were they to be transmitted across a network or the Internet, image compression is desirable.

Alternatively, 8-bit color images can be used to hide information. In 8-bit color images, (such as GIF files), each pixel is represented as a single byte. Each pixel merely points to a color index table, or palette, with 256 possible colors. The pixel's value, then, is between 0 and 255[5]. The image software merely needs to paint the indicated color on the screen at the selected pixel position. If using an 8-bit image as the cover-image, many steganography experts recommend using images featuring 256 shades of gray as the palette, for reasons that will become apparent. Grey-scale images are preferred because the shades change gradually from byte to byte. This increases the image's ability to hide information.

3. Methodology

In this approach to transmit the data over internet we divide the whole procedure into two parts or layers. The message is represented by M and a key selected K for the encryption scheme application. Now the encoded algorithm is applied over the data and the text of 128 bit is obtained. As the text to be transmitted is too large it can be compressed. Thus the compression is applied to generate a 48 bit key.

The second layer obtains this 48 bit as input to the cover image which is called container image. Container image is chosen so as to have minimum distortion in carrier after insertion. Thus procedure includes the following phases:

a. Encryption

To perform encryption the message (M) and key (k) are selected and the whole text is divided into group of 128 bits. Various operations applied are modulus, transform as well as substitution cipher in order to obtain cipher text.

The steps followed for algorithm are:

Generate the bit pattern of message string and partition into group of 128bits.

Shift the bits right to apply transformation and modulus operation.

Rotate the bits towards left position.

Obtain cipher text by substitution and transformation of ciphers.

Thus here they obtained result is cipher text which is of 128 bits and the next is to compress this.

b. Compression

As to reduce the amount of data to be transmitted the compression technique is applied at layer 1 where the 128 bits generated are compressed to 48 bits. The algorithm for conversion is given as: Generate the ASCII code array of cipher text obtained.

Calculate intensity of occurrence of each code.

Arrange the values in descending order of occurrences.

Assign the bit pattern accordingly to each code.

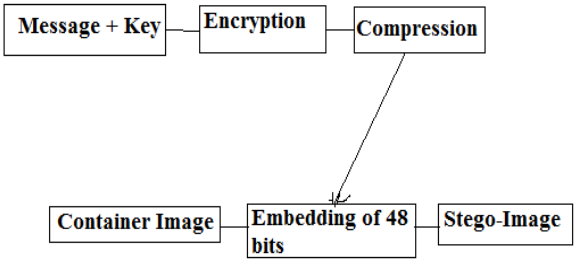

Fig. 1. Block Diagram of Dual Layer Approach

Thus the layer 1 performs the refining and scaling task and in layer 2 the embedding of data is performed. At the receiver end reverse processes are followed to retrieve the text.

c. Embedding

Here we are embedding the encoded text into the image file which will thus send over internet. Container image file is processed so as to find the maximum intensity value pixels. Now as a pixel is surrounded by maximum eight pixels and most intense pixel will have all eight surrounding pixels of different color values leading to distinction. Thus such pixels are selected and used for replacement. As in 24 bit image the 3 bits are stored in each pixel so for 48 bits we require 16 pixels of different intensity. Thus descending order of intensity is formulated and top 16 pixels are selected to replace their LSB with encoded key pattern. The algorithm followed is:

Embed_data(image)
1. [Find maximum intensity pixels of colour image]

Intensity[]= getpixel(colorvalue (x,y))
2. [Apply sorting to obtain array values in descending order.]

Sort(intensity[])
3. For (pixel=1 to 16 of intensity[]) do
3.1 Generate bit pattern of pixel.
3.2 XOR 48 bit key and LSB of pixel.
4. Return Stego image.

Now as stated earlier each pixel is checked for the color intensity of surrounding pixels thus a grid layout is obtained which divide whole image into the small part ions to scan for each pixel up to three layers.

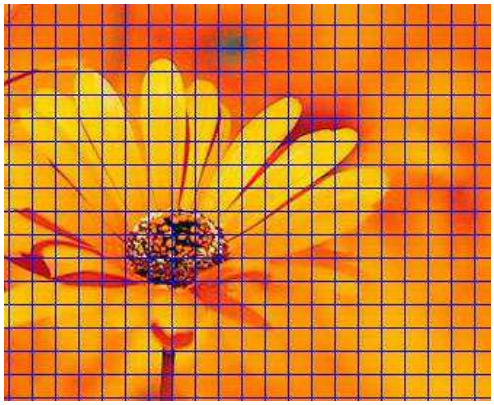

Fig. 2. Grid Layout Of Image For Pixel Selection

Thus after the calculation the bit pattern of pixels is obtained. However the one small block thus generated for image is checked once for the pixel and this will reduce the time needed for the embedding of data.

Now after the replacement of encoded data bits with pixel‘s LSB there is need to set up the colour values of replaced pixels so as to maintain the originality of image. The change in pixel value depends upon the RGB colour values of the pixels such that the range varies from 0,0,0 from blach to the 255,255,255 for white colour. Now as LSB is the most presice bit so the slight changes with the value of LSB doesn‘t make much effect on the image colour. Thus color value doesn‘t change much in LSB replacement.

d. Data Retrieval

Now at the receiver side the data is extracted from the image by the process called data retrieval. Firstly the embedded bits are calculated from image and then the decryption is performed through the key. This includes performing the following steps:

Image is processed to find out the location of high intensity pixels across intensity array.

Now all the pixels are formulated so as to retrieve the bit pattern of each container pixel.

LSB of each bit pattern is extracted and the key is applied for decryption in order to retrieve the text.

The difference between color intensity of container image and stego image is not perceptible by human eye and thus reduces the chances of interception.

4. Experimental Results

This approach is successfully implemented in visual C#.net.Firstly container image is selected on the basis of the RGB color values. Selection of 24 bit color image is made however in this approach. Now if for example the message to hide is ―hello covert‖ in a 14 x 8 image fragment. The message is 96 bits in length(12 characters with 8 bit each) and image has 112 pixels . Thus after compression the key data obtained is 48 bits. As the image is 24 bit thus a pixel contains 3 bytes. So pixels needed to store 48 bits are 16 . Thus the whole image must have 16*N+C pixels where N is bits embedded and C is remain no. of pixels.

Consider the following images that are container image used for hiding data:

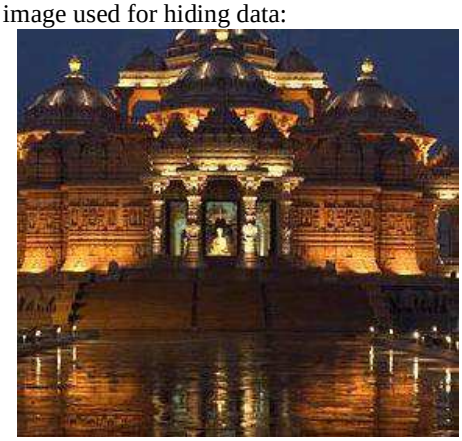

Fig.3 Container image with size 567 KB

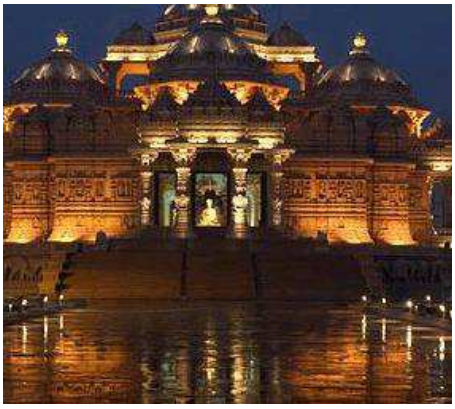

Fig. 4. Stego Image with size 768 KB and 66 KB data

The color variation in two images is negligible. The security can be considered two layers encryption layer and hiding layer. Thus if in any way attacker knows data is hidden in image data is still encrypted and can‘t be easily encoded. In this algorithm data is scattered in the whole image and thus extraction without knowledge of intensity parameters is almost negligible.

5. Conclusion

A new approach is proposed to embed information within an container image. This method will be expected to spread hidden information within encrypted image data randomly based on the secret key before transmission. Thus, this information appears to be nothing out of the usual and should be available to the receiver safely. However the information can be extracted at receiver‘s side by again finding high intensity pixels and extraction being performed. The time consumption is less and is easily negotiable. However very small distortion of image can occur after inserting text but effect is negligible. It is the better way to provide security to data transfer over internet.

6. References

[1] A. Sinha, K. Singh, "A technique for image encryption using digital signature," Source: Optics Communications, vol.218, no. 4, 2003, pp.229-234.

[2] A.F. Al-Husainy. Mohammed, "Image Encryption Using Genetic Algorithm," Journal of Information Technology, Vol. 5, No. 3, 2006, PP.516-519.

[3] MAB. Younes, A. Jantan, "Image Encryption Using Block-Based Transformation Algorithm," IAENG International Journal of Computer Science, Vol. 35, Issue. 1, 2008, pp.15-23.

[4] EE. Kisik Chang, J. Changho, L. Sangjin, "High Quality Perceptual Steganographic Techniques," Springer, Vol. 2939, 2004, pp.518-531.

[5] Elke, Fraz, "Steganography preserving statistical properties," proceeding of the 5th internationally Workshop on information Hiding, Noordwijkerhout, The Netherlands, October 2003, LNCS 2578, Springer 2005, pp. 278-294.

[6] W.-K. Chen, G. C. Kessler, "Steganography: Hiding Data Within Data," An edited version of this paper with the title "Hiding Data in Data," originally appeared in the April 2002 issue of *Windows & .NET Magazine*. September 2001.

[7] H. El-din H. Ahmed, M. K. Hamdy, and O. S. Farag Allah, "Encryption quality analysis of the RC5 block cipher algorithm for digital images," Optical Engineering, Vol. 45, Issue 10107003.