IMAGE BLOCK-BASED STEGANOGRAPHY AND ENCRYPTION SYSTEM FOR BUSINESS APPLICATIONS

First A. Dr. T.V. Ananthan, Department of Computer Science and Engineering, E-mail: tvananthan@rediffmail.com,[1];
Second B. R. Mohan Lakshmanan, Department of Computer Science and Engineering, mohanlskshmanan@yahoo.com [2],
Dr. M.G.R Educational and Research Institute, Maduravoyal, Chennai, India- 600095

Abstract

In this modern era, communication between peers need secrecy of user data and ensuring the implementation of genuine transactions is at the expense of providing reliability. In this paper, we have proposed a steganographic technique to provide security for user accounts through encryption and image bock based steganography. Encryption of user data and hiding it in a cover image using Image blocks to generate a undetectable stego image will provide privacy. Because the attackers cannot identify the secret message hidden inside the cover image with random block size to generate a stego image ensure legitimacy in business process.

Keywords— Steganography, Encryption, Network Security.

Introduction

In cryptography, encryption is the process of transforming information (referred to as plaintext) using an algorithm (called a cipher) to make it unreadable to anyone except those possessing special knowledge, usually referred to as a key. The result of the process is encrypted information (in cryptography, referred to as cipher text). The reverse process, i.e. to make the encrypted information readable again is referred to as decryption [10]. Encryption, by itself, can protect the confidentiality of messages, but other techniques are still needed to protect the integrity and authenticity of a message such as verification of a message authentication code (MAC) or a digital signature. Standards and cryptographic software and hardware to perform encryption are widely available, but successfully using encryption to ensure security may be a challenging problem.

Encryption and Steganography are analogous to each other because both provide the luxury of hiding the actual data and make it visible to the legally authorized entities alone. Encrypting a data does take different forms with different algorithms but the key idea is ensuring safety of the data transmitted. Since data passes through insecure networks is viable to attacks by hackers. It is therefore necessary to safeguard the data from being theft by the illegitimate sources. The techniques in use have rather served as a method to protect the secrecy behind the data encrypted and elsewhere is subject to compromise.

Related Work

In general, "computer security" often refers to addressing three important aspects of a computer related system-confidentiality, integrity and availability [7, 8]. To do so, steganography laid hands to provide strong mechanism to hide vulnerable data passing the insecure networks. Steganography is the art of concealing the existence of information within unobjectionable carriers [2]. Steganography, in an essence disguise a message and make it invisible thus hiding the fact that a message is hidden behind. An invisible message will not be suspicious whereas an encrypted message will be [1].

By far the most popular and frequently used Steganographic method is Least Significant Bit insertion method [5]. It works by embedding message bits as the LSBs of the randomly selecting pixels in the image. Later R. Rana and Er. Dheerendra Singh have proposed a LSB insertion technique which uses pre determined random pixel and segmentation of image for concealing the message [4]. We have some other interesting works of steganography including the one done by Miroslav Dobiscek, where the content is encrypted with one key and decrypted with several keys in which the relative entropy between encrypt key and one specific decrypt key corresponds to the amount of information [9]. The popularity of LSB insertion technique has been due to the false belief that the modification of pixel value by 1 in randomly selected pixels is not detectable but now that myth has expired shortly[3].

Proposed Methodology

The proposed system has two levels of data concealment in which the encryption algorithm is used to encrypt the secret message which can be user id and password. The encrypted data obtained as a result of encryption will be embedded in a cover image predefined using block-based pixel insertion technique.

A. Encryption of the Secret Message

The encryption of the secret message is done using bit-map images. The data to hidden in the cover image is a valid user data which may user id or password vulnerable to attack through e-shopping, e-banking and e-ticketing, etc.

Assuming a user data of length 20 characters has to be hidden in a cover image. This is done in our proposed method using the encryption algorithm as follows:

Encryption Algorithm:

Input: SM, SMarray[], r1[],r2[]
Output: Chararr[j], SMascii[], Secret Message Array in ASCII Format.
BEGIN:
Step 1:
For each digit d of SM do
SMarray[i] = ConvertTobinary(4)
end for
For each char C in SM do
SMarray[i] = ConvertTobinary(5)
end for
Step 2:
For every integer i in SMarray[] do
If r1[i] and r2[i] both zero then
SMarray[i] = concatenate (”00”, SMarray[i])
Else r1[i] and r2[i] both one then
SMarray[i] = concatenate (”11”, inverse (SMarray[i]))
Else if r1[i] = 0 and r2[i] =1 then
SMarray[i] = (SMarray[i]) – 0001
SMarray[i] = concatenate (“01”, SMarray[i])
Else if r1[i] = 1 and r2[1]= 0 then
SMarray[i] = SMarray[i] + 0001
SMarray[i] = concatenate (“10”, SMarray[i])
end for
For each char i in SMarray[] do
j = i
Chararr[j]
If r1= 0 then
SMarray[i] = concatenate (r1, SMarray[i])
Else if r1 =1 then
SMarray[i] = concatenate (r1, SMarray[i])
End
Step 3:
For each 6-bit value in SMarray[i] do
SMdec[i] = ConvertToDecimal (SMarray[i])
end for
Step 4:
For each integer i in SMdec[i] do
SMrad64 [i] = ConverTtoRadix64 (SMdec[i])
end for
Step 5:
For each char i in SMrad64[i] do
SMascii [i] = ConvertToAScii (SMrad64[i])
end for

The encryption of the secret message is carried out based on the input given by the user. The input can be string of characters or an n- digit number or a combination of both. The encryption of the secret message uses radix64 conversions as claimed in [6]. Fig 1 depicts the encryption of an integer using the algorithm. Here the integer to be encrypted is converted to 4 bit binary number and along with two binary digits it forms 6-bit value which went through a series of conversions to obtain a ASCII value. This ASCII value will be used to embed the secret message in the cover image to produce a stego image.

B. Embedding the Encrypted data using Image Block – based Embedding algorithm

The encrypted data will be embedded in the chosen cover image by splitting the cover image into various blocks. The encrypted data is a ASCII value which is between 0 and 255. So to embed a value in particular block in the cover image, we will chose to change the particular pixel in a chosen block and all the bits in the pixel will be set to its one’s complement value.

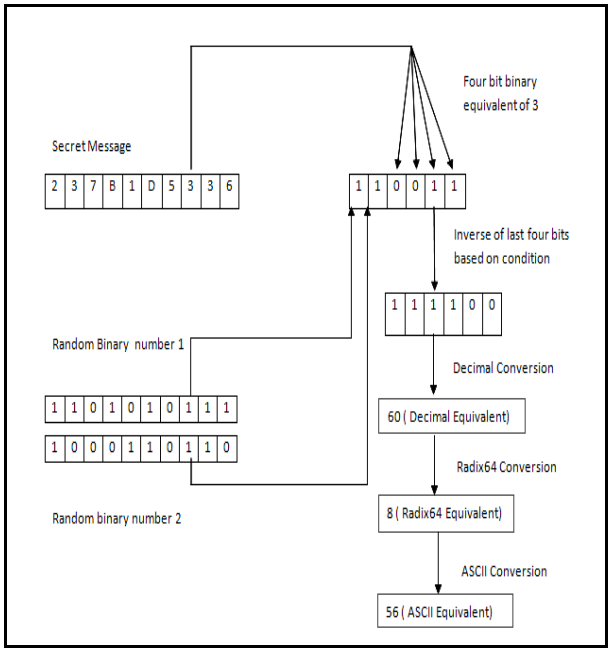

Figure 1- Encryption of an Integer

The block-based embedding of secret message is done using the following order considering there are n ASCII values in the Secret Message generated by the encryption algorithm then n blocks will be created in the image. For each block, 256 pixels is needed to embed a ASCII value. As a whole, n x 256 pixel image is required to embed the secret message.

Once the cover image is ready, the secret message embedded will be hidden in the reverse order such that nth block contains 1st ASCII value in the Secret message array SMascii[]. Similarly (n-1)th block of the cover image will contain the ASCII value in the 2nd value of the SMascii[] array. Finally the 1st block will contain the nth ASCII value in the array. Fig 2 shows the encryption of an alphabet using the encryption algorithm.

For each block, 256 pixel is needed for embedding the data and the first 16 x 16 = 256 pixel range will form the first block in the cover image. The next 16 x 16 pixel range will be the embedding block for the next secret message in the array. Hence to embed a secret message of length n we need blocks each with 256 pixels.

Consider a cover image of size 256 x 256 is selected for embedding the message, then a total of 65536 pixels is available for insertion. But we will use only n pixels for inserting the secret message which will be a very less pronounced change in the actual image and will not be identifiable to the human eye. This makes the task of detecting the message hidden inside more unlikely to occur. The Stego image generated is less variable in revealing the statistical properties to the actual cover image.

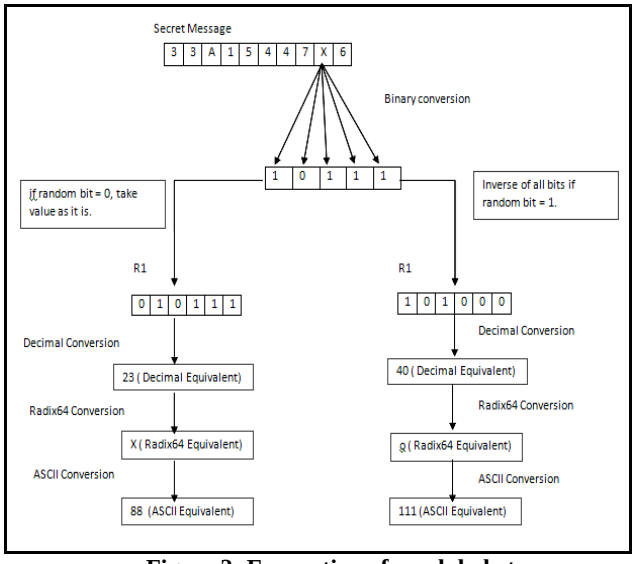

Figure 2- Encryption of an alphabet

The embedding algorithm used for storing the secret message in the cover image to generate the stego image is as follows:

Embedding Algorithm:
Input: SMascii[], Cover image, n
Output: Stego image
Steps:

1. Store the blocksize n of the image
2. Store the message size to be hidden in the image
3. Store the values of character array Chararr[] in the image in specified space
4. For each value i =1 in SMascii[i] do
5. While the data left to embed do
6. Identify the nth block in the cover image
7. Get the first value of the secret message array
8. With each block
9. Scan to find the pixel equal to the value of the secr message array obtained for the fixed block size
10. Set one's complement of all the bits of the pixel
11. end with
12. Move to the (n-1)th block of the image
13. Get the next value of the secret message array
14. Until (n>0)
15. end while
16. end for
17. end

C. Extracting the embedded message

Figure 3 - Cover Image

The extraction of the embedded secret ASCII code from the cover image is done using the extraction algorithm. The cover image as shown in fig.3 is given as input along with the stego image to the algorithm and the algorithm scans the stego image block-wise to detect the change in pixel value as

with the predefined cover image. If there is a change, the pixel position will be copied into the Secret Message array which is the output of this algorithm as follows:

Extracting Algorithm:
Input: Stego image, Cover image
Output: SMascii[], chararr[]
Steps:
1. Retrieve blocksize stored in the image
2. Retrieve size of message n hidden in the image
3. Retrieve char array values to be stored in chararr[]
4. For i =1 to n block in Stego image
5. For j = 1 to blocksize
6. Compare pixels in nth block of cover image with pixel in nth block of stego image
7. If one's complement bit insertion performed in a pixel then
8. Copy the pixel range as a ASCII value to the Secret message array in (n- i)th position
9. else if move to the next block
10. end if
11. end for
12. end for

D. Decryption of the encrypted data

The Secret message extracted from the extracting algorithm has all the values in ASCII equivalents and it will be given as input to unearth the hidden data and is shown below:

Decryption Algorithm:
Input: SMascii[], chararr[]
Output: Secret Message
BEGIN:
Step 1:
For i =1 to sizeof (SMascii[]) in SMascii[] do
SMrad64[i] = ConvertTorad64(SMascii[i])
end for
Step 2:
For each char i in SMrad64[i] do
SMdec[i] = ConvertTodecimal(SMrad64[i])
end for
Step 3:
For each integer i in SMdec[] do
SMbin[i] = ConvertToBinary(SMdec[i])
end for
Step 4:
For each index i in SMbin[] do
j = chararr[i]
If index i is not equal to chararr[i] then
If bit b1[i] and b2[i] both zero in SMbin[i] then
SMbin[i] = SMbin[i](b3b4b5b6)
Else if b1[i] and b2[i] both one then
SMbin[i] = Inverse (SMbin[i](b3b4b5b6))
Else if b1[i] = 0 and b2[i] =1 then
SMbin[i] = SMbin[i](b3b4b5b6) + 0001
Else if b1[i] = 1 and b2[1]= 0 then
SMbin[i] = SMbin[i](b3b4b5b6)-0001
end if
Else
If bit b1 = zero in SMbin[j] then
SMbin[i] = SMbin[j](b2b3b4b5b6)
Else if b1 = one in SMbin[j] then
SMbin[i] = Inverse(SMbin[j](b2b3b4b5b6)
end if
end if
end for
Step 5:
For every binary value in SMbin[i] do
SMArray[] = ConvertToDecimal (SMbin[i])
end for

E. Results and Discussions

The algorithm used for embedding the secret message was applied on a 211 x 239 pixel bmp image shown in fig 3. It is 24bpp image. The image used as cover image in fig 3 has the capacity hold 211 x 239 x 3 = 151287 bytes of data (i.e) 50429 pixels available. We have reserved two bytes cover image for storing the block size used for embedding the secret message and two bytes for storing the character array positions and one byte for the size of the message. Consider the block size to be 128 pixels then with the cover image of size 211 x 239 maximum of 391 pixels can be modified to stored the secret message.

Table 1- Bits Per Pixel in Block 1

PIXEL NO	PIXEL BITS
1	101110110101010101010111
5	101110110101010101010111
67	101110110101010101010111
113	101110110101010101010111
124	111111110111111101010101

The blocks in Stego Image has the data inserted in it is not visible to the human eye and is evident from the values generated using the cover image and stego image. The block with different sizes with multiple pixels per block modified is shown in the fig 4.

Table 1 shows the bits per pixel values in a cover image block. The bits in each pixel will remain same for both the cover image and stego image after embedding the secret message unless there is pixel modification for a particular pixel correspond--ding to the insertion array is shown in Table 2.

Table 2 shown below will display the pixel values generated in the stego image after performing the stuffing of one's complement of the bits in that pixel. This one's complement operation will result in inverse of the bits of that position which will be equivalent to the respective ASCII positional values of the encrypted data obtained from the encryption algorithm. This value change is subject to take place for a pixel within a block of the selected range and hence it is very much unlikely to affect the statistical properties of the cover image. Figure 4 shows the three blocks of different size with one's complement insertion at various positions in each block depending on the encoding value. It is evident that this image has not shown any visible change that is identifiable by a naked eye.

Table 2 – A bitwise pixel modification in Block1

PIXEL NO	PIXEL BITS
113	101110110101010101010111 (Before Insertion of 1's complement)
113	010001001010101010101000 (After insertion of 1's complement)

Even if in case the steganalyst identify this as a stego image, it is not possible to get the sequence of blocks considered for regenerating the cover image from the stego image and in turn the encrypted data. Hence this technique is far more secure and safe from cryptanalytic as well as steganaly-tic attacks and it finds to be a better prospect for implementing secured business applications. Since the stego image does not show appreciable difference in statistical properties of the cover image and stego image which is evident from the Histogram analysis of both the cover image and stego image shown in fig 5.

It is less certain for an attacker to detect the presence of a hidden message within the cover image, since the extraction of the secret message is done with the necessary information such as block size and the size of the secret message to be embedded in the cover image. So if an attacker came to know that there is a message embedded in the image and it is worthy to be extracted, it is not a simple task as there is a need to have multiple details of the data including size of the block and kind of data hidden which is necessary for extraction. Otherwise, the attacker may not be able to find the hidden message. The greater complexity in hiding the data the lesser the chance of unveiling the secret message which makes it foolproof for unrevealing the message.

Figure 4 - Stego Images of varying block size with a pixel modification in each block

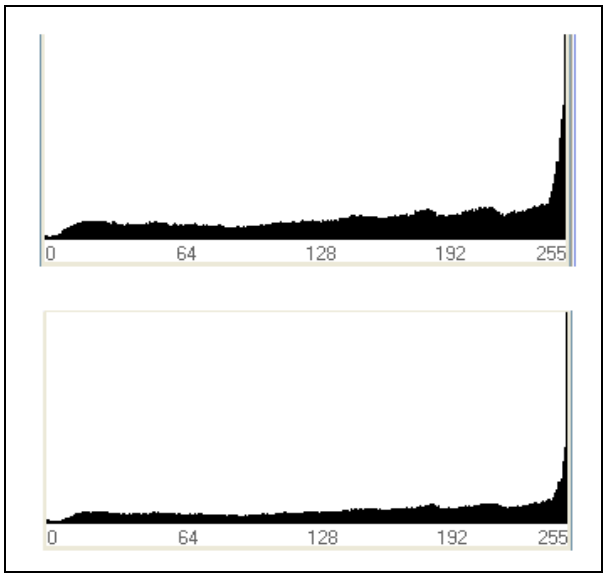

Figure 5- Histogram analysis of Cover Image and Stego Image

Conclusion

This work on image block –based embedding and encryption of the secret message has used steganography to a consumable way to hide data and to transfer it securely over an unsecured network. This method of steganography ensures the data sent through networks s free from the attack by intermediate hackers .There are two levels of security one with multiple level conversion based encryption for varied data and block based embedding scheme of secret message

that ensure safe transfer of data used in business applications over a network. Thus data theft and peer repudiation were checked by this technique of encryption and steganography.

References

[1] Neil F. Johnson, Sushil Jajodia, George Mason University(1996), "Exploring Steganography: Seeing theUnseen", IEEE Computers, February 1998, pp. 26-34.

[2] Lisa M. Marvel, Charles G. Boncelet, and Charles T. Retter (1998), "Reliable Blind Information Hiding for Images", 2nd Information Hiding Workshop, 1998.

[3] Ritesh Rana, Er. Dheerendra Singh, "Steganography-Concealing Messages in Images Using LSB Replacement Technique with Pre-Determined Random Pixel and Segmentation of Image", IJCSC, 2010

[4] W. Bender, D. Gruhl, N. Morimoto, A. Lu (1996),"Techniques for Data Hiding" IBM Systems Journal, 35, Nos 31996.

[5] Ross Anderson, Roger Needham, Adi Shamir, "The Steganographic File System", 2nd Information Hiding Workshop, 1998.

[6] William Stallings, "Cryptography And Network Security", p. No 475 -480, Fourth Edition.

[7] G.R.Blakley, Twenty years of cryptography in the open literature, Security and Privacy 1999, Proceedings of the IEEE Symposium, 9 -12 May, 1999.

[8] W. K Chen, Scott Sutherland, "An introduction of Cryp--tography", MSTP MATH WORKSHOP, 2005.

[9] M Dobiscek, "Extended Steganographic System", 8th International Conference on Electrical Engineering, FEE CTU 2004, Poster 04.

[10] Wikipedia , "http://en.wikipedia.org/wiki/Encryption".

Biographies

FIRST Dr.T.V.ANANTHAN, Professor in the Department of Computer Science & Engineering and Information Technology in Dr MGR Educational and Research Institute, Chennai. He has got 20 years of experience in teaching and projects. He has published six papers in International Journals and Three National Conferences. Author can be reached by mail through tvananthan@rediffmail.com.

SECOND R. MOHAN LAKSHMANAN, M .Tech in Computer Science and Engineering from DR. MGR Educational and Research Institute, Maduravoyal, Chennai. He has published research papers in two National Conferences. Author can be reached by mail through mohanlakshmanan@yahoo.com.