[Volume 3 issue 1 January 2015]
Page No.824-828 ISSN :2320-7167

INTERNATIONAL JOURNAL OF MATHEMATICS AND COMPUTER RESEARCH

A Steganography Method to Embed Text in Image without Change Structure of Image

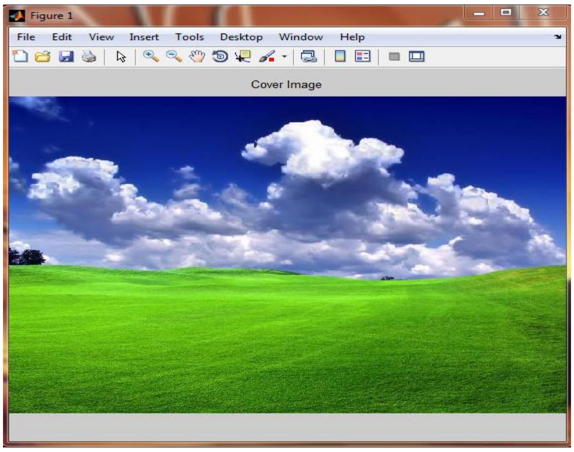
Dr. Saad Abdual azize AL_ani, Bilal Sadeq Obaid Obaid
Associate Prof. Computer Science
Department Computer Sciences and Informatics Researcher
Al_Mamon University College
saadabdualazize@yahoo.com

Abstract
Steganography is the process of hiding one file inside another file that neither identify the meaning of the embedded object, nor even recognize its existence. Current trends favor using digital image files as the cover file to hide another digital file that contains the secret message or information depending on the value of pixel for digital image.

Key word : image ,encryption ,decryption , embedded

Introduction

The simplest and the most common steganographic technique is the Least Significant Bit embedding (LSB). The premise here is that changes to the least significant bit will be masked by noise which is commonly present in digital images. Actually, in the case of color images, there is even more room for hiding messages because each pixel is a triple of red, green, and blue. Again, replacing two or more least significant bits of each pixel increases the capacity of the scheme but at the same time the risk of making statistically detectable changes also increases. Therefore, it is important to study the security of each specific steganographic technique and argue why it is secure. Also, the simple least significant bit encoding might introduce detectable changes under certain circumstances.
The security of the transformation of hidden data can be obtained by two ways: encryption and steganography[5]. A combination of the two techniques can be used to increase the data security [1]. In encryption, the message is changed in such a way so that no data can be disclosed if it is received by an attacker [2]. Whereas in steganography [3], the secret message is embedded into an image often called cover image, and then sent to the receiver who extracts the secret message from the cover message [4]. When the secret message is embedded into cover image it is called a stego-image. The visibility of this image should not be distinguishable from the cover image, so that it almost becomes impossible for the attacker to discover any embedded message



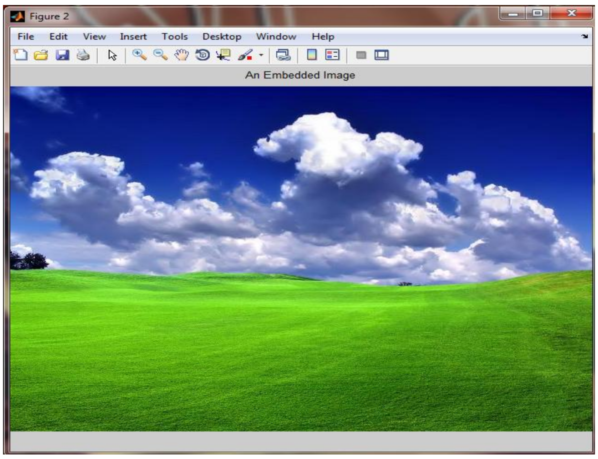


Figure.2: An Embedded Image

Proposed Algorithm

This algorithm is based on convert the plain text to binary ,and convert each pixel in cover image to four part each 4 bit

Embedding Algorithm

a- Plain text

1. Divide the plain text to 8 characters for each part.
2. Convert each character to 6 bits using ,get 48 bits.
3. Group of 6 bits in s-table to obtain 4 bit ,and totally 32 bits ,the outer two bits of each group become row of s-table ,the middle four bits become column of s-table , Each 4 bit call M1,M2,…..M8
4. Each RGB pixel contains three numbers.
5. Convert each number to 8 bits, the output 24 bits.
6. Divide them to 4 bits and each part can call p1, p2…….p6.
7. Choose randomly four parts.
8. Each part filled as row in array (4,4) .
9. For each row ,exclusive –or all bits in that row to get one bit for r1,r2,r3,r4
10. Arrange the bit from r1 to r4 to be R1
11. For each column ,exclusive –or to all bits in that column to get one bit forc1,c2,c3,c4
12. Arrange the bit from c1 to c4 to beC1
13. Choose another randomly for part of p's
14. Repeat step from 6 to 9 to get R2 and C2.
15. Apply from step 2 to step 11 on 8 pixels
16. At end obtain array (8,4) each row contain value of pixel with its R's and C's .
17. Compare each M from plain with contain of array if found , Store
    A- Location of M five bits as.
    B- Three bit number of pixel.
    C- Two bits value of R's or C's.

18. Each character converts to 5 bits , totally 40 bit , grouped to 4 each group contain 10 bit , convert each group to decimal
19. Apply RSA algorithm on each group get 4 cipher text
20. Send the cipher text as text file with image.

Implementation

Image

Read the image, each image contains a pixel, each pixel contains RGB color

Pixel i

P1 = first 4 bits from R-color   P2 = second 4 bits from R -color
P3 = first 4 bits from G-color   P4 = second 4 bits from G-color
P5 = first 4 bits from B-color   P6 = second 4 bits from B-color

Pixel 1

18 = 00010010 , p1 = 0001 p2= 0010

20 = 00010100 , p3 = 0001 p4 = 0100
17 = 00010001 , p5 = 0001 p6 = 0001

Randomly, build table1 from p1, p2, p3, p4 and table2 from p1, p4, p5, p6

P1	**0**	**0**	**0**	**1**
P2	**0**	**0**	**1**	**0**
P3	**0**	**0**	**0**	**1**
P4	**0**	**1**	**0**	**0**

Table 1

P1	**0**	**0**	**0**	**1**
P4	**0**	**1**	**0**	**0**
P5	**0**	**0**	**0**	**1**
P6	**0**	**0**	**0**	**1**

table 2

R1 = 1111 C1 = 0110 R2 = 1111 C2 = 0101

Pixel2
14 = 00001110
19 = 00010011
15 = 00001111

P1	0	0	0	0
P4	0	0	1	1
P5	0	0	0	0
P6	1	1	1	1

P1	0	0	0	0
P2	1	1	1	0
P3	0	0	0	1
P4	0	0	1	1

R1 = 0110
R2 = 0000

C1 = 1100
C2= 1100

P	R1 00		C1 01		R2 10		C2 11	
P1/000	1111	15	0110	6	1111	15	0101	5
P2/001	0110	6	1100	12	0000	0	1100	12
P3/010	0111	7	1110	14	0101	5	1100	12
P4/011	0010	2	1000	8	0010	2	0100	4
P5/001	1011	11	1101	13	1111	15	0110	6
P6/101	0010	2	1111	15	1010	10	1111	15
P7/110	0010	2	1011	11	1010	10	1111	15
P8/111	0101	5	1100	12	1011	11	1011	11

Table 1: value of each pixel and weight of R's and C's

The message: ENCRYPTION
Convert each character to 6 bits, obtain the row and column from table 2, and take the value of intersection as shown in s-table:
E=000100 r=00 = 0 c = 0010 = 2

S(0,2) = 15 = P1R1 = 00000

SEQ	Char	Weight	Row & column	Intersection value	Vale from table1	weight

1	E	000100	0,2	15	P1R1	00000
2	N	001101	1,6	6	P2R1	00100
3	C	000010	0,1	8	P4C1	01101
4	R	010001	1,8	5	P3R2	01010
5	Y	011000	1,12	2	P4R2	01110
6	P	001111	1,7	11	P8C2	11111
7	T	010011	1,9	12	P2C1	00101
8	I	001000	0,4	10	P7R2	11010

Table 2: weight each character, its value from s-table and its value from table1

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	8	15	4	10	2	4	6	10	3	5	8	2	11	15	7
1	9	7	3	13	10	1	6	11	5	12	7	11	2	6	2	1
2	4	12	1	13	7	9	10	9	15	11	14	0	8	2	5	13

S-table

Merge each two binary numbers and convert to decimal
P1r1=00000 and P2R1 = 00100 = 0000000100 convert to decimal = 4
Using RSA algorithm for p= 117 ,q = 19 e = 11 d = 1139

Number	binary	number	Binary	Both	Decimal	RSA
P1R1	00000	P2R1	00100	0000000100	4	1726
P4C1	01101	P3R2	01010	0110101010	426	810
P4R2	01110	P8C2	11111	0111011111	239	1451
P2C1	00101	P7R2	11010	0010111010	186	972

Table 3: arrange each 2 numbers

Decrypt the output 4, 170, 239, 186 using RSA algorithm
The cipher text = 1726 , 810 , 1451 , 972
Write the cipher text in text file, send it with the image

Decryption

Convert the cipher text to plain text , convert to 8 bits ,divide to two part ,4 bits each ,part 1 represent number of pixel , part 2 represent R's or C's, take the value ,search in table3 column I by column , get the intersection (0,2),convert to:

Cipher Text	Plain text	10 bits	Part1	Intersection Value in table1	It's Value in table1	f part2	Intersection Value in table1	It's Value in table1
1726	4	0000000100	00000	P1R1	15	00100	P2R1	6
810	426	0110101010	01101	P4C1	8	01010	P3R2	5
1451	239	0111011111	01110	P4R2	2	11111	P8C2	11

972	186	0010111010	00101	P2C1	12	11010	P7R2	10

Table 4

It's Value in table1	**Row & Column**	**Binary number**	**Char**
15	**0,2**	**000100**	**E**
6	**1,6**	**001101**	**N**
8	**0,1**	**000010**	**C**
5	**1,8**	**010001**	**R**
2	**1,12**	**011000**	**Y**
11	**1,7**	**001111**	**P**
12	**1,9**	**010011**	**T**
10	**0,4**	**001000**	**I**

Conclusion

This paper proposed a new method of hiding data in the cover image. This algorithm is based on converting character to 6 bit, by using b-table compressed to 4 bit and from another side used similarly to value of cover pixel. The receiver gets value of location encrypted by RSA algorithm. There is no data embedded in the cover image.

References

[1] W. Bender, D. Gruhl, N. Morimoto, A. Lu, "Techniques for data hiding", IBM Systems Journal, Vol. 35, Issue 3-4, 1996, pp. 313-336.

[2] F. Petitcolas, R. Anderson and M. Kuhn, "Information Hiding-A Survey", Proceedings of the IEEE, Special Issue on Protection of Multimedia Content, Vol. 87, Issue 7, July 1999, pp. 1062-1078.

[3] N.F. Johnson and S. Jajodia, "Exploring steganography: seeing the unseen", Computer, Vol. 31, Isuue 2,February 1998, pp. 26-34.

[4] K. Stefan and A. Fabien, "Information hiding techniques for steganography and digital watermarking",Artech House Books, December 2000

[5] William Stallings; Cryptography and Network Security: Principals and Practice, Prentice Hall international, Inc.; 2002.

[6] Eric Cole,"Hiding in Plain Sight: Steganography and the Art of Covert Communication"