Volume 3, Issue 5, May-2016, pp. 226-232 ISSN (O): 2349-7084

International Journal of Computer Engineering In Research Trends

Available online at: www.ijcert.org

Suggested method for hiding secret data in cover image depending on the Connected Component Labeling algorithm

Atheer Hussein Zyara

Abstract:- the encryption and concealment of important areas to maintain the confidentiality of data , Where the proposed method is working on the coordination between the confidential data and the cover image and that is through combining between the encryption process and the concealment process (Creating the appropriate environment to conceal confidential data encoded in a special way at specific locations According to certain conditions) , Where the proposed method is divided into three phases , First stage : converting confidential data to a binary system , And then dividing them to two matrices , 1's matrix that contains 1's with 0 between each frequent two groups from 1's , and 0's matrix that contains 0's with 1 between each frequent two groups from 0's . Second stage: converting the cover image to labeled image and thus determining the locations of pixels that its value 1 and the locations of pixels that its value 0 .Third stage : hiding 1's matrix and 0's matrix in the specific locations , And thus obtaining embedded image . Then sending the embedded image, And four keys (the start storage location, and length for 1's matrix. the start storage location, and length for 0's matrix) to the recipient to compare it with original image (unembedded image) for extracting the secret data . This method is characterized by flexibility where can be applied on image types and data of different, and the durability because the concealment locations be random semi. To illustrate the proposed method has been tested on the cover image of type the gray sacle , and textual data , also using the metrics (PSNR) and (MSE) To test the efficiency of the proposed method And that by using the language of MATLAB .

Keywords – proposed algorithm, LSB technique, Connected components labeling technique, Encrypting and decrypting data, Hiding and extracting data .

1. INTRODUCTION

Maintaining the confidentiality of the data of important areas, which was and still is a lot of researchers are trying to find the best ways, the techniques used to maintain the confidentiality of data are divided into two parts. Encryption techniques that are working to change the overall shape of confidential data , while The second section is the concealment techniques and working to remove the shading and doubts about the existence of secret data , This was the technique applied since ancient times and With the rapid development in the field of computers have been exploited better . To increase the security of confidential data can be a combination between encryption technique and concealment technique , When someone tries to gain access to confidential data on him revealed concealment , If enable, it will face data is unregularly and this will remove doubt in existence of the hidden data .[1][2]

2. ENCRYPTION & DECRYPTION

Encryption is one of the used techniques in maintaining the security of confidential data, and so by changing the overall shape of confidential data, where appearing on that it illogical Information, There are two ways for this technique. symmetric which needs to only one key , which is used in the encryption and decryption , and asymmetric and that need to be two keys , public key , which is used in the encryption and the private key is used in decryption . The type of symmetric was used in this research. [7]

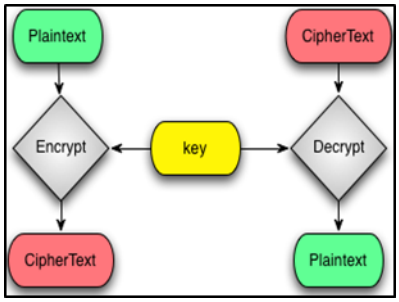

Fig.1 Confidential data encryption and decryption

3. STEGANOGRAPHY

One of the techniques that are used in maintaining the security of data and that through hiding of confidential data in another file, so that we can send confidential data to the other party without raising suspicions. There are three ways to hiding: [3][4][5]

3.1. Pure :

In this method Concealment and retrieving does not require a secret key.

Hiding: cover*secreat_data >> Embedded cover
Retrieving: Embedded cover >>secret_data

3.2. by using Secret Key :

In this method of concealment and retrieval needs for secret key .

Hiding: cover*secreat_data *key >> Embeded_cover
Retrieving: Embeded_cover * key >> secret*data

3.3. by using Public secret Key and Private secret Key :

The public key is used to hiding a data either the private key is used to data extraction.

Hiding : cover*secreat_data *key1 >> Embeded_cover
Retrieving : Embeded_cover * key2 >> data_secret

Where:

key : the secret key to hide and extract data

key1: the puplic key

key2: the private key

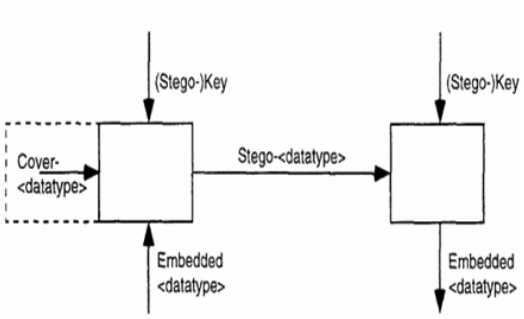

Fig.2. Hiding confidential data and extracting confidential data

4. CONNECTED COMPONENTS LABELING

Connected components labeling is technique that using to scanning the image and its pixels and that depending on connectivity pixels (It describes the relationship between more than pixels, and there are conditions that must be fulfilled to be there connected pixels , the conditions are the pixel brightness and spatial adjacency) ,all connected pixels are similar pixel in intensity values and are in some way connected with each other. Pixel is labeled with a gray level or a color according to a component it was assigned. [2][8]

Pixel connectivity

The process is starting from the peaks and spreads in all the image based on the connectivity of the pixels , and defines the pixels are connected to other pixels.

4.1. 4-connected

4-connected pixels are neighboring to each pixel that touches any of their edges. (Connected horizontally and vertically). For pixel coordinates. Each pixel contain the coordinates (x 1,y) or (x,y 1) is connected to pixel at (x,y) .

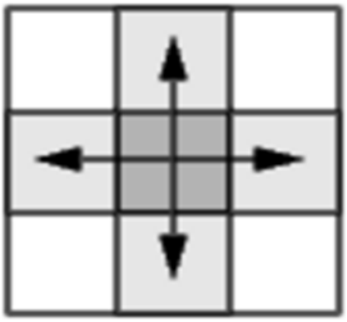

Fig.3 4-connected

4.2. 8-connected

8-connected pixels are neighboring to each pixel that touching one of their edges or corners. 8-connected pixels are connected vertically, horizontally, diagonally. In addition to 4-connected pixels, every pixel with coordinate (x 1,y 1) or (x 1,y 1) is connected to pixel in (x,y) .

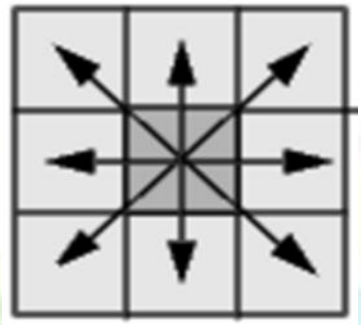

Fig.4 8-connected

5. LEAST SIGNIFICANT BIT (LSB)

The lowest bit in a series of numbers in binary system, the LSB is located at the far right of a string. where The one's bit of a byte is used to encode the hidden information as (location 8 , location 7 , location 6) .[3][4][6]

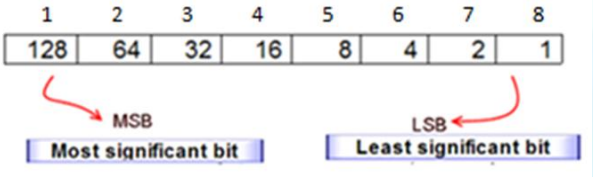

Fig.5. Least significant bit (LSB)

6. THE PROPOSE METHOD

There are several stages:

6.1. Converting the secret data

in this stage secret data are converting to binary system , where the binary system consists of 1's and 0's , so we will depend on sequence for (1's) and sequence for (0's) . Collect them in a single matrix, then dividing it into two matrices. 1's Matrix which contains the sequence (1) with (0) between each two sets of the (1's) , 0's Matrix which contains the sequence (0) with (1) between each two sets of the (0's) , where the matrix (1) is complementary to the matrix (0) and vice versa .

6.2. Converting the cover image

At this stage, the cover image is converted to a labeled image. Thus determining the concealment locations, where indexed locations are using to hiding the 1's matrix, While other locations are using for storage 0's matrix. [2][8]

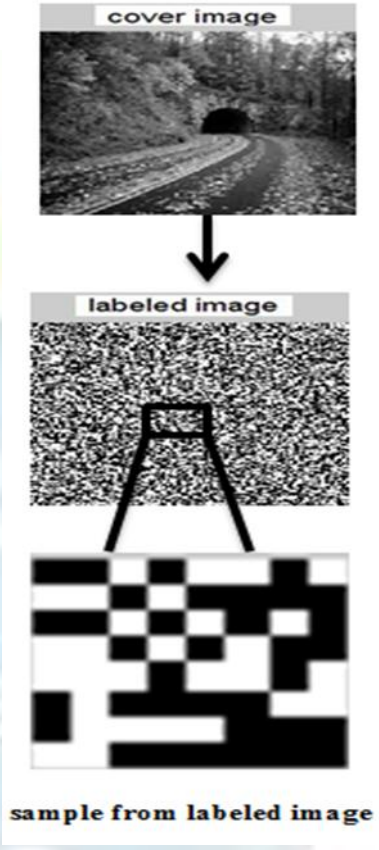

Fig.6 converting a cover image for labeled image

6-3 hiding the secret data in cover image

Depending on stage first and second stage, hiding matrix (1's) and matrix (0's) in cover image (depending on technique LSB), and so depending on the identified locations to concealment from labeled image. Thus we get the embedded image. [3][4][6]

6-4 sending of embedded image

Sending of embedded image with four keys (the start storage location , and length for 1's matrix . the start storage location , and length for 0's matrix) The recipient.

6-5 extracting the secret data

After receiving the embedded image, it will be compared (As is the case in the cover-ups process) with

the original image and depending on received secret keys with embedded image, will be extracted confidential data. .[3][4][6]

Algorithm & Example (1) Encrypting and hiding confidential data

Input: secret data (S_data) , cover image (C_img) .

Output : embedded image (E_img) , four keys (first location (FL_1) to store 1's matrix , and matrix length (ML_1) . first location (FL_0) to store 0's matrix , and matrix length (ML_0))

Step_1 : Encryption of confidential information

1-Reading of secret data (S_data)

Suppose we have a sample from secret data is " God " .

2-Converting S_data to a decimal system (S_dec)

D [71 111 100]

3-Converting S_dec to a binary system (S_bin)

S_bin: [1000111;1101111;1100100]

4-Putting S_bin in Matrix one-dimensional (S_bin_m)

S_bin_m: [100011111011111100100]

5-Separating S_bin_m (by private programming instructions that designed for this purpose) into mat_1(matrix of 1's that contains 1's with 0 between all frequent group from 1's) , and mat_0 (matrix of 0's that contains 0's with 1 between all frequent group from 0's)

mat_1 :[10111110111111010]

mat_0 : [00010100100]

Step_2 : converting of cover image into labeled image

1-eading of cover image (C_img)

suppose we have a sample from cover image is 8*5

C_img=

91	91	91	91	91
91	4	0	0	91
91	0	127	5	91
3	0	127	0	1
1	127	127	127	2
0	2	0	1	0
3	0	43	0	2
0	43	43	43	0

matrix of two dimensions (1) Cover image before embedding (C_img) , N=8 , m=5

2- Convert C_img into new image (N_img) Composed of 0 or 1 (but not binary) , and that by this rule : (if value of pixel is odd number , transform it for 1 , Otherwise transform it for 0)

If Pixel_value mod 2 == 0

Pixel_value is equal to 0

Else

Pixel_value is equal to 1

N_img=

1	1	1	1	1
1	0	0	0	1
1	0	1	1	1
1	0	1	0	1
1	1	1	1	0
0	0	0	1	0
1	0	1	0	0
0	1	1	1	0

matrix of two dimensions (2) N_img

3- Convert N_img into labeled image (L_img)

L_img=

1	1	1	1	1
1	0	0	0	1
1	0	1	0	1
1	0	1	0	1
1	1	1	1	0
0	0	0	1	0
2	0	3	0	0
0	3	3	3	0

Matrix of two dimensions (3) L_img

Step_3 : determining the concealment locations

1-determining the objects for L_img

2-put pixel locations for every objects in matrix one-dimension (OPL_mat) , while other pixel locations, which are not within the objects , put it in matrix one-dimension (not_OPL_mat) . Thus, we will get on FL_1, ML_1, FL_0, and ML_0

Matrix of object pixel locations OPL_mat :

[1,2,3,4,5,9,13,17,19,20,21,25,

29,30,33,34,35,36,7,16,23,24,32]

Matrix of one-dimension (4)

OPL_mat

Matrix of not object pixel locations not_OPL_mat : (matrix of one-dimension)

[6,8,10,11,12,14,15,18,22,

26,27,28,,31,37,38,39,40]

matrix of one-dimension (5) not_OPL_mat

The secret keys

FL_1 : 15 (or from any location from OPL_mat matrix)

ML_1: 18

FL_0 : 6 (or from any location from not_OPL_mat matrix)

ML_0: 10

Step_4 : hiding of secret data (S_data) in cover image (C_img)

hiding mat_1 in cover image depending on OPL_mat values , and hiding mat_0 in cover image depending on not_OPL_mat values . and that by using LSB technique. Thus, we will get on E_img

91	91	90	91	91
90	0	0	0	90
91	1	127	0	91
3	0	126	1	1
1	127	127	127	2
0	3	1	0	0
3	0	43	0	2
0	43	43	43	0

E_img=Matrix of two dimensions (6) Cover image after embedding (E_img)

Step_5 : sending E_img , FL_1, ML_1, FL_0, and ML_0 into recipient .

Flowchart (1) the process of concealment

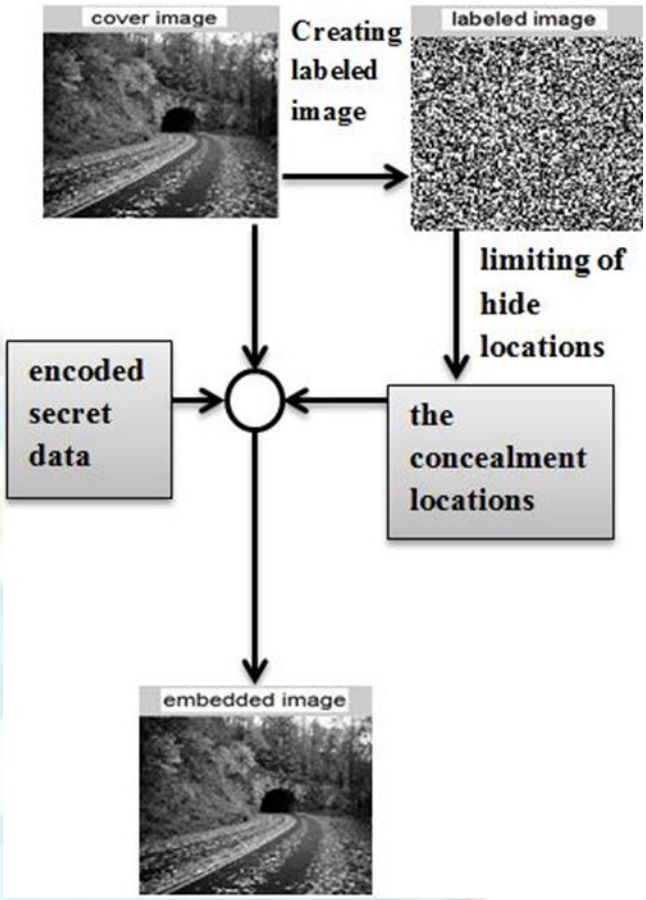

Fig.7 Hiding of secret data

Algorithm & Example (2) extracting the secret data from stego_image

Input : C_img , E_img , FL_1, ML_1,FL_0 , ML_0 .

Output : secret data (S_data)

Step_1: the same step_2 in algorithm (1)

Converting of cover image into labeled image

Step_2: the same step_3 in algorithm (1)

determining the concealment locations

Step_3: Extracting the secret data 1's Matrex , and 0's Matrix from the embedded image (E_img)

1-Extract confidential data (mat_1) of the embedded image, depending on (FL_1,ML_1) and the specific locations of the labeled image for cover image .

mat_1 :[10111110111111010]

2-Extract confidential data (mat_0) of the embedded image, depending on (FL_0,ML_0) and the specific locations of the labeled image for cover image .

mat_0 : [00010100100]

Step_4: decrypting the confidential information

1-converting of confidential data (mat_1,mat_0) by using some instructions into S_bin_m

S_bin_m: [100011111011111100100]

2-reshaping S_bin_m into Matrix two-dimensional (S_bin) where , N > 0 , and M=7

S_bin: [1000111;1101111;1100100]

3-Converting S_bin to a decimal system (S_dec)

D : [71 111 100]

4-Converting S_dec to a characters (S_data)

" God "

Flowchart (2) the process of Extraction

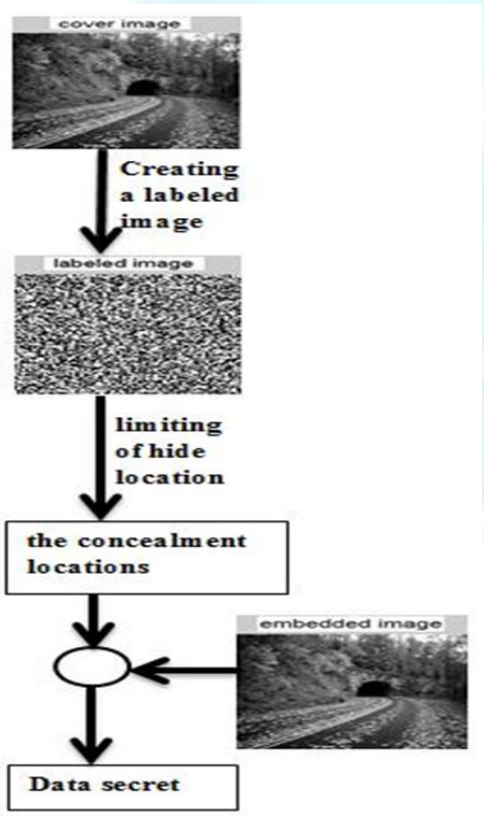

Fig.8 Hiding of secret data

7. THE RESULTS

To clarify the proposed method has been tested on the gray scale image, and textual data and so by using Efficiency standards MSE and PSNR.

The locations of storage start (LSS)

The location which starting from it the storage for mat_1 in OPL_mat , and mat_0 in not_OPL_mat as (11, 200,45, 718,....,etc) and which will give the same results .

Table 1. Display the results of test

LSS , Image size, connective type		Text length by bits	hiding location of bit from byte	PSNR	MSE
LSS with Image size of cover (143*143)	4-connected	560	8	67.0391	0.0130
		2009	8	61.5310	0.0461
		4018	8	58.5207	0.0921
		560	7	59.0704	0.0812
		2009	7	53.5483	0.2895
		4018	7	50.5483	0.5776
		560	6	52.9875	0.3294
		2009	6	47.5248	1.1588
		4018	6	44.5159	2.3168
	8-connected	560	8	67.0391	0.0130
		2009	8	61.5310	0.0461
		4018	8	58.5207	0.0921
		560	7	59.0704	0.0812
		2009	7	53.5483	0.2895
		4018	7	50.5483	0.5776
		560	6	52.9875	0.3294
		2009	6	47.5248	1.1588
		4018	6	44.5159	2.3168

8. DISCUSS THE RESULTS

Through studying the results table, figures, and algorithms above show us:

PSNR: Directly proportional with size of image .inversely with size of data and the hiding location of bits in byte.

MSE: Directly proportional with size of data and the hiding location of bits in byte. Inversely with size of image.

LSS: the Values of PSNR and MSE are same with changing the values of LSS.

Image size: the values of PSNR and MSE are changing with changing image size.

Image type: the values of PSNR and MSE are doing change with changing image type.

Connective type: when using (4-connected or 8-connected) the values of PSNR and MSE are not changing.

9. CONCLUSIONS

1-It is impossible to detect hidden data from the embedded image without the original image and the concealment keys.

2-the Concealment sites are random completely. Reason they are selected on the basis of the labeled image.

3-Hiding large amounts of data without affecting the shape of the cover image.

4-Values of efficiency standards (PSNR and MSE) Is excellent.

5-Retrieve the hidden data completely.

6-Coordination between the Connected Component Labeling technique and separating secret data works to reduce the hidden data and the increase in the random concealment locations therefore increasing proposed method security.

10. REFERENCES

[1] F. I. Alam, F. K. Bappee and F. U. A. Khondker , " An Investigation into Encrypted Message Hiding Through Images Using LSB", International Journal of Engineering Science and Technology (IJEST), Vol. 3 No. 2, pp. 948-960, Feb 2011.

[2] Grana C, Borghesani D, Cucchiara R: Optimized block-based connected components labeling with decision trees. IEEE Trans Image Process 2010, 19(6):1596-1609.

[3] P. K. Singh, and R.K. Aggrawal., " Enhancement of LSB Based Steganography for Hiding Image in Audio", IJCSE International Journal on Computer Science and Engineering, Vol. 02, No. 05, pp. 1652-1658, 2010.

[4] Gutte, R.S., Chincholkar Y.D., (2012), "Comparison of Steganography at One LSB and Two LSB Positions", International Journal of Computer Applications, Vol. 49, No.11.

[5] H.A.Jalab, A.A Zaidan, B.B Zaidan, "New Design for Information Hiding With in Steganography Using Distortion Techniques", International Journal of Engineering and Technology (IJET)), Vol. 2, No. 1, Feb (2010).

[6] Singh, S. and Agarwal, G., (2010), "Use of Image to Secure Text Message With the Help of LSB Replacement", International Journal of Applied Engineering Research, Dindigul Vol. 1, No.1.

[7] Wayner, Peter (2009). Disappearing cryptography 3rd Edition: information hiding: steganography & watermarking. Amsterdam: MK/Morgan Kaufmann Publishers.

[8]He L, Chao Y, Suzuki K: A run-based two-scan labeling algorithm. IEEE Trans Image Process 2008, 17(5):749-756.

