Data Concealment Technique in Background Segmentation of Color Image Based Secure Data Transmission System

Khamael A. Al-Phalahi
Department of Computer Science, College of Science, University of Al-Nahrain, Al-Jaderyia, Baghdad-Iraq.

Abstract

Image based steganography is the most popular method for data concealment. In this paper a new approach for data concealment in background segments of color image called JSEG, **(J-image segmentation)** is presented. A criterion to local windows in the class-map results of the "J-image", is applied in which high and low values correspond to possible region boundaries and region centers, respectively then a region growing method is used to segment the color image to regions based on the multi-scale J-images. The adopted system combines the effect of two methods to enhance the security of the data. **System encrypts the data with a crypto algorithm** and then **embeds the encrypted text in a segmentation background of color image file** and then merged background with other regions segmentation. The (increase in PSNR) leads to provide high secret communications, so the attacker cannot notices the difference between the stego-image and the original cover.

1. Introduction

The growing possibilities of modern communications need the special means of security especially on computer network. The network security is becoming more important as the amount of data being exchanged on the Internet is increasing. Security requirements are necessary both at the final user level and at the enterprise level, especially since the massive utilization of personal computers, networks, and the Internet with its global availability. Throughout time, computational security needs have been focused on different features: secrecy or confidentiality, identification, verification, non repudiation, integrity control and availability ***[1].***

There are a number of ways for securing data. ***One is Cryptography*** It is the practical art of converting messages or data into a different form, such that no-one can read them without having access to the ***'key'.*** The message may be converted using a 'code' (in which case each character or group of characters is substituted by an alternative one), or a ***'cypher' or 'cipher'*** (in which case the message as a whole is converted, rather than individual characters).The ***second method is steganography***, where the secret message is embedded in image, thus the existence of message is unknown. Computer based steganography allows changes to be made to what are known as digital carriers such as images or sounds. Digital images, videos, sound files, and other computer files that can be used as "covers" or carriers to hide secret messages **[2].**

After embedding a secret message into the cover-image, a so-called ***stegoimage*** is obtained .The basic model of steganography consists of Carrier, Message, Embedding algorithm and ***Stego key***. Carrier is also known as a ***cover-object***, which embeds the message and serves to hide its presence. The suitable carriers that can be used as cover object is Image files such as ***bmp, gif*** and ***jpg,*** where they can be both ***color*** and ***grayscale [1]*** .

Message is the data that the sender wishes to remain it confidential. It can be ***plain text***, ***cipher text***, other image, or anything that can be embedded in a hit stream such as a copyright mark, a covert communication, or a serial number. Cryptographic systems are generically classified along three independent dimensions ***[2]:***

1. *Methodology for transforming plain text to cipher text.*

All encryption algorithms are based on two general principles: substitution, in which each element in the plaintext is mapped into another element, and transposition, in which elements in the plaintext are rearranged. The fundamental requirement is that no information be lost.

2. Methodology for number of keys used.

If both sender and receiver use the same key, the system is referred to as symmetric, single-key, secret-key, or conventional encryption. If the sender and receiver each use a different key, the system is referred to as symmetric, two keys, or public-key encryption.

3. Methodology for processing plain text.

A block cipher processes the input one block of elements at a time, producing an output block for each input block. A stream cipher processes the input elements continuously, producing output one element at a time, as it goes along. The proposed algorithm uses a substitution cipher method. It is a symmetric key algorithm using the technique of stream cipher ***[2].*** The *combination of cryptography and steganography are shown in **Fig.(1).***

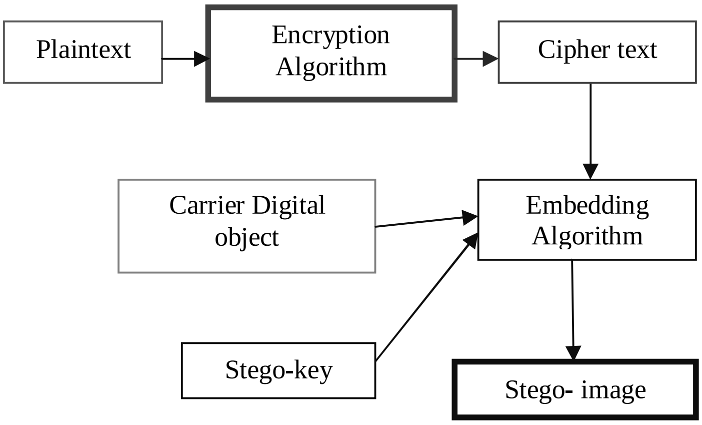


Fig.(1) combination of cryptography and steganography.

2.Encryption algorithm

DES (***Data Encryption Standard)*** algorithm is a careful and complex combination of two fundamental building blocks of encryption: *substitution* and *transposition*. It's a symmetric block cipher, operating on 64-bit blocks using a 56-bit key that encrypts data in blocks of 64 bits. ***The input*** to the algorithm is a 64-bit block of plaintext and ***the output*** from the algorithm is a 64-bit block of ciphertext after 16 rounds of identical operations. The key length is 56 bits by stripping off the 8 parity bits, ignoring every eighth bit from the given 64-bit key. As with any block encryption scheme, there are two inputs to the encryption function: the 64-bit plaintext to be encrypted and the 56-bit key. The basic building block of DES is a suitable combination of permutation and substitution on the plaintext block (16 times**)** ***[3].***

Description of the DES Algorithm

The plaintext block X is first transposed under the initial permutation IP, giving $X_0 = IP(X) = (L_0,R_0)$. After passing through 16 rounds of permutation, XORs and substitutions, it is transposed under the inverse permutation IP^{-1} to generate the cipher text block Y. If $X_i = (L_i,R_i)$ denotes the result of the *i*th round encryption, then we have

$$Li=R_{i-1} \quad\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$
$$Ri=L_{i-1} \oplus f(R_{i-1},K_i) \quad\ldots\ldots\ldots\ldots\ldots (2)$$

The *i*th round encryption of DES algorithm is shown in Fig.(2) The block diagram for computing the f(*R*,*K*)-function is shown in Fig.(3). The decryption process can be derived from the encryption terms as follows:

$$R_{i-1}=L_i \quad\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3)$$
$$L_{i-1}=R_i \oplus f(R_{i-1}, K_i) = R_i \oplus f(L_i, K_i) \ldots\ldots (4)$$

If the output of the *i*th round encryption be $L_i \| R_i$, then the corresponding input to the (16–*i*)th round decryption is $R_i \| L_i$. The input to the first round decryption is equal to the 32-bit swap of the output of the 16th round encryption process. The output of the first round decryption is $L_{15}\|R_{15}$, which is the 32-bit swap of the input to the 16th round of encryption ***[3].***

The key is 64 bits long, but in fact it can be any 56-bit number. The user can change the key at will any time there is uncertainty about the security of the old key. ***Key Schedule*** the 64-bit input key is initially reduced to a 56-bit key by ignoring every eighth bit. These ignored 8 bits, *k*8, *k*16, *k*24, *k*32, *k*40, *k*48, *k*56, *k*64 are used as a parity check to ensure that each byte is of old parity and no errors have entered the key. After the 56-bit key was extracted, they are divided into two 28-bit halves and loaded into two working registers ***[3].***

The halves in registers are shifted left either one or two positions, depending on the round. After being shifted, the halves of 56 bits (C_i, D_i), $1 \leq i \leq 16$, are used as the key input to the next iteration. These halves are concatenated in the ordered set and serve as input to the Permuted Choice 2 which produces a 48-biy key output. Thus, a different 48-bit key is generated for each round of DES. These 48-bit keys, $K_1, K_2, . . . ,K_{16}$, are used for encryption at each round in the order f from K_1 through K_{16} ***[4]*** .

There are two different S-box mappings S_0 and S_1:

S_0 if a_i=0 ,then S-box S_0 transforms the input 4-bit data <u>d</u>.

S_1 if a_i=1, then S-box S_1 transform the input 4-bit data <u>d</u>.

The nibble a_i= (a_0, a_1,a_2,a_3) determines which of the 2^4=16 possible S-box combinations is used to transform the 16bits of data r in Step L_2 as specified by the next equation **[5]**.

<u>d</u>=(d_0,d_1,d_2,d_3) $\longrightarrow$ (Sa_0 (d), Sa_1 (d), Sa_2 (d), Sa_3 (d)).

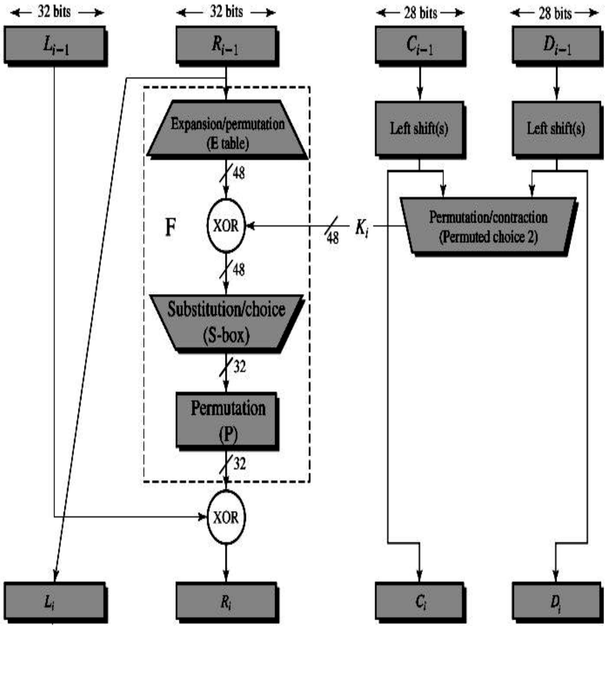


Fig.(2) Single round encryption of DES algorithm.

<u>Example (1)</u>

For the input (101110) to S_5-box, denote as S_5^{10}(0111), the first and last bits combine to form 10, which corresponds to the row 2 (actually third row) of S_5. The middle 4 bits combine to form 0111, which corresponds to the column 7 (actually the eighth column) of the same S5-box. Thus, the entry under row 2, column 7 of S5-box is using [**Table (3.6) S-boxes**, ***[4]]***computed as:

$S_5^{10}(0111) = S_5^{2}\ (7)$ = 8 *(*hexadecimal*)* = 1000 *(*binary*)*

Thus, the value of 1000 is substituted for 101110. That is, the four-bit output 1000 from S5 is substituted for the six-bit input 101110 to S_5

<u>Example (2)</u> Suppose the 64-bit plaintext is *X* = 3570e2f1ba4682c7, key input is *K* = 581fbc94d3a452ea including 8 parity bits. The first two-round keys are, respectively, K_1=27a169e58dda and K_2= da91ddd76748 . For the purpose of demonstration, the DES encryption aims to limit the first two rounds only.

1. The plaintext *X* splits into two blocks (L_0, R_0) using (**Table (1.a)**) IP such that L_0 = ae1ba189 and R_0 = dc1f10f4.
2. The 32-bit R_0 is expanded to the 48-biy E (R_0) such that E (R_0) = 6f80fe8a17a9.
3. The key-dependent function Γ_i is computed by XORing E (R_0) with the first round key K_1, such that in ***(Eq.2)*** Γ_1 = E (R_0) ⊕ K_1= ***4821976f9a73***
4. This 48-bit Ω1 is first divided into eight six-bit blocks, and then fed into eight S*i*-boxes. The output Ω_1 resulting from the S-box substitution phase is computed as Ω_1 = **a1ec961c**. Using (**Table (2.d)**) the permuted values of Ω1 are P(Ω_1) = 2ba1536c. Modulo-2 addition of P(Ω_1) with L_0 becomes R_1= P(Ω_1) ⊕ L_0= ***85baf2e5***
5. Since $L_1 = R_0$, such that in ***Eq.(1)*** this gives L_1 = dc1f10f4. Consider next the second-round encryption. Expanding R_1 with the aid of **(Table (2.c))** yields E (R_1) = c0bdf57a570b. XORing E (R_1) with K_2 produces
   Ω2= E(R_1) ⊕ K_2= ***1a2c28ade043***
6. The substitution operations with S-boxes yields the 32-bit output Ω_2 such that Ω_2 ***1ebcebdf*** .Using **(Table (2.d))**, the permutation P(Ω_2) becomes *P(Ω_2)*= ***5f3e39f7***. Thus, the right-half output R_2 after round two is computed as ***[4]***:

R_2 = P (Ω_2) ⊕ L_1= ***83212903***

7. The left-half output L_2 after round two is immediately obtained as
L_2= R_1 = ***85baf2e5***

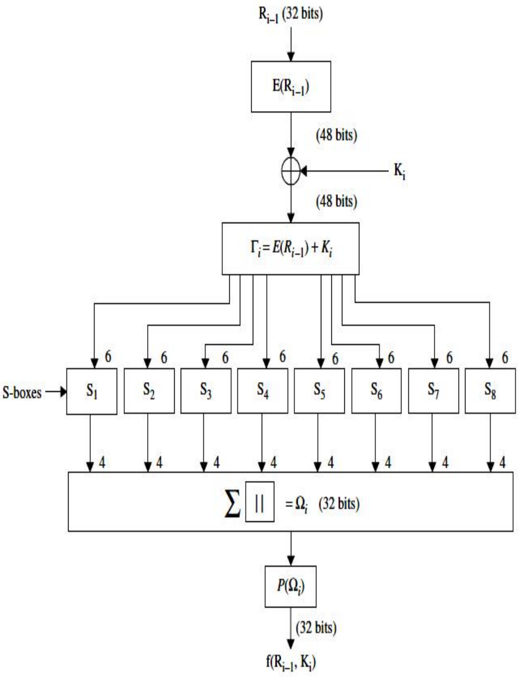


Fig.(3) Computation of the f-function.

8. Concatenation of R_2 with L_2 is called the pre output block in our two-round cipher system. Thus, the output of the DES algorithm at the end of the second roundbecomes the cipher text Y using (**Table (1.b)):**
Y = $IP^{-1}(R_2||L_2)$= ***d7698224283e0aea***

Table (1).

(a) Initial Permutation (IP)							
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

(b) Inverse Initial Permutation (IP^1)							
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Table(2)

(c) Expansion Permutation (E)							
	32	1	2	3	4	5	
	4	5	6	7	8	9	
	8	9	10	11	12	13	
	12	13	14	15	16	17	
	16	17	18	19	20	21	
	20	21	22	23	24	25	
	24	25	26	27	28	29	
	28	29	30	31	32	1	
(d) Permutation Function (P)							
16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	

Security is a big challenge for computer users. Businessmen, professionals, and home users all have some important data that they want to secure from others. In this technique we have the software for data encryption and then embed the cipher text in background of color image based on ***color image***

segmentation. This algorithm combines the effect of these two methods to enhance the security of the data. The method encrypts the data with a DES algorithm and then embeds the encrypted text in background of color image file. This method improves the security of the data by embedding the encrypted text and not the plain text in an image. This technique will satisfy four requirements which we must require for secure data transmission. These are ‘Confidentiality', or Message Content Security, Integrity of Message Content, Authentication, Security in an open system ***[5].***

3-1 Criterion for Segmentation

Colors in the image are coarsely quantized without significantly degrading the color quality. The purpose is to extract a few representing colors that can be used to differentiate neighboring regions in the image. A good color quantization is important to the segmentation process. The quantized colors are assigned labels. A color class is the set of image pixels quantized to the same color ***[5].***

- The class-map can be viewed as a set of spatial data points located in a 2-D plane. The value of each point is the image pixel position, a 2-D vector *(x,y)*.
- These data points have been classified and each point is assigned a label. Before proceeding further, let us first consider the measure *J* defined as follows. Let *Z* be the set of all *N* data points in the class-map. let z=(x,y),

$z \in Z$, and *m* be the mean,

$$m = \frac{1}{N}\sum_{z\in Z} z \quad \text{...}(5)$$

Suppose *Z* is classified into *C* classes, *Zi* , i=1,…, C Let m_i be the mean of the *Ni* data points of class *Zi*,

$$m_i = \frac{1}{N_i}\sum_{z\in Z_i} z \quad \text{..}(6)$$

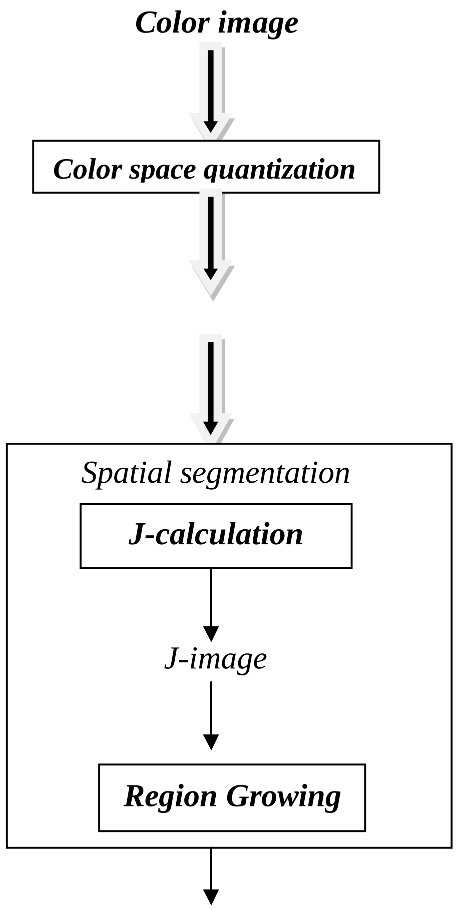


Fig.(4) Schematic of the JSEG algorithm. Let.

$$S_T = \sum_{z\in Z}\|z-m\|^2 \quad \text{..}(7)$$

and

$$S_W = \sum_{i=1}^{C} S_i = \sum_{i=1}^{C}\sum_{z\in Z_i}\|z-m_i\|^2 \quad \text{...................................}(8)$$

The measure *J* is defined as

$$J = \frac{S_B}{S_W} = \frac{(S_T - S_W)}{S_W} \quad \text{................................}(9)$$

- It essentially measures the distances between differ ent classes *SB* over the distances between the members within each class S_W .A higher value of *J* indicates that the classes are more separated from each other and the members within each class are closer to each other, and vice versa ***[6].***

3-2 Spatial Segmentation Algorithm

The characteristics of the *J*-images allow us to use a region-growing method to segment the image. Fig.(5) shows a flow-chart of the steps in our spatial segmentation algorithm. Consider the original image as one initial region the algorithm starts segment all the regions in the image at an initial large scale ***[5].***

i-***Valley Determination***

At the beginning, a set of small initial areas are determined to be the bases for region growing. These areas have the lowest local *J* values and are called valleys. In general, finding the best set of valleys in a region is a non-trivial problem. The following simple heuristics have provided good results in the experiments:

a. Calculate the average and the standard deviation of the local *J* values in the region, denoted by and respectively.

b. Set a threshold T_J at

$$T_J = m_J + a s_J \quad \text{...} (10)$$

Pixels with local *J* values less than T_J are considered as candidate valley points. Connect the candidate valley points based on the 4-connectivity and obtain candidate valleys.

c. If a candidate valley has a size larger than the minimum size scale,it is determined to be a valley.

d. *a* is chosen from the set of parameter values [–0.6, –0.4, –0.2, 0, 0.2, 0.4] which gives the most number of valleys ***[6].***

ii-Valley Growing

The new regions are then grown from the valleys. It is slow to grow the valleys pixel by pixel. A faster approach is used in the implementation:

a. Remove “holes” in the valleys.

b. Average the local *J* values in the remaining unsegmented part of the region and connect pixels below the average to form growing areas. If a growing area is adjacent to one and only one valley, it is assigned to that valley.

c. Calculate local *J* values for the remaining pixels at the next smaller scale to more accurately locate the boundaries. Repeat steps 2 that is shown in Fig.(5).

d. Grow the remaining pixels one by one at the smallest scale. Unclassified pixels at the valley boundaries are stored in a buffer. Each time, the pixel with the minimum local *J* value is assigned to its adjacent “valley” and the buffer is updated till all the pixels are classified ***[6].***

3-3 Region Merge

After region growing, an initial segmentation of the image is obtained. It often has over-segmented regions. These regions are merged based on their color similarity. The quantized colors are naturally color histogram bins. The color histogram features for each region are extracted and the distances between these features can be calculated. Since the colors are very coarsely quantized, in our algorithm it is assumed that there are no correlations between the quantized colors. Therefore, a Euclidean distance measure is applied directly. First, distances between two neighboring regions are calculated and stored in a distance table. The pair of regions with the minimum distance are merged together. ***[6].***

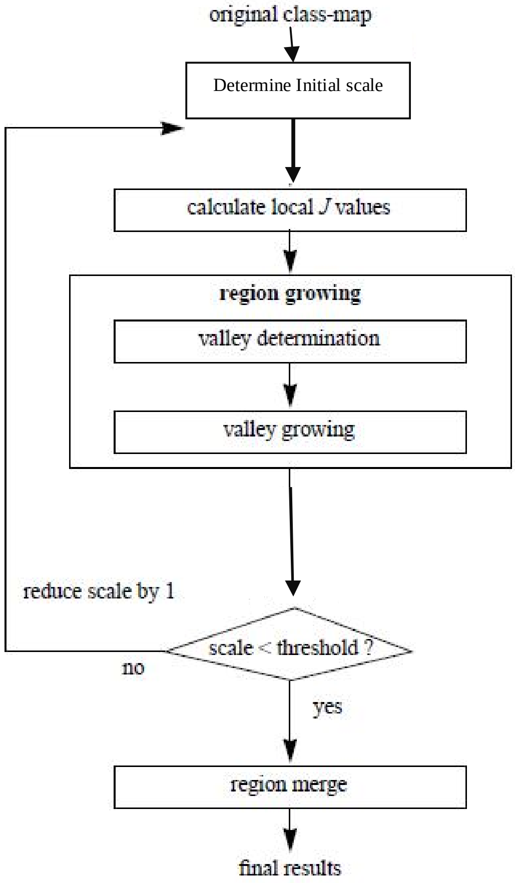
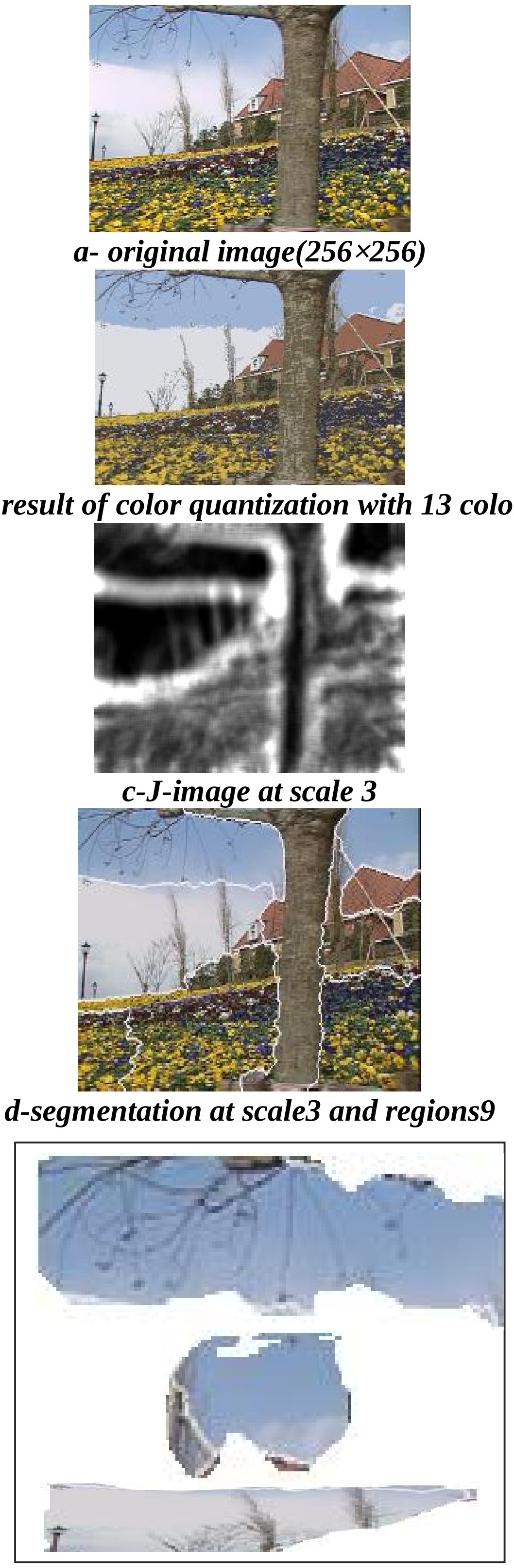


Fig.(5) Block diagram of the steps in spatial segmentation.



a- original image(256×256)

b-result of color quantization with 13 color

c-J-image at scale 3

d-segmentation at scale3 and regions9

e- result of segment background of image

Fig.(6) a- original image b-result of color quantization with 13 colors c-J-image at scale 3 d-result after segmentation at scale 3 and regions 9 e- result of segment background of image.

4-Concealment data in digital image

The most common approaches to information hiding in images are; Least significant bit (LSB) insertion ,Masking and filtering techniques, Algorithms and transformations ***[7].***

The least significant bit insertion method is probably the most well known image steganography technique. It is a common, simple approach to embed information in a graphical image file. This is the most common method used in this the data to be hidden is inserted into the least significant bits of the pixel information In digital, images are represented with the numerical values of each pixel where the value represents the color and intensity of the pixel. In 24 bit image we can embed 3 bits in each pixel while in 8-bit we can embed only 1 bit in each pixel. To hide an image in the LSBs of each byte of the 24- bit image, one can store 3 bits in each pixel. A1024×768 image has the potential to hide a total of 2,359,296 bits of information ***Steganography methods namely Steo1bit, Stego2bits, Stego3bits,***
Stego4bits, StegoColourCycle [7].
Stego1Bit method involves utilizing a single least significant bit of one of the RGB bytes of a 24-bit image for message concealment. As the color value is not changed much, it will not considerably alter the visual appearance of color and image.
Stego2Bits method involves utilizing two least significant bits of one of the RGB bytes of a 24-bit image for message concealment. Although the capacity of data storage is 2 times improved than Stego1Bit, the resulting image is degraded than Stego1Bit ***[8].***

For example (3):

The letter A can be hidden in three pixels. Assume the original three pixels are represented by the three 24-bit words below:
(00100111 11101001 11001000) (00100111 11001000 11101001) (11001000 00100111 11101001)
The binary value for the letter A is (10000011). Inserting the binary value of A into the three pixels, starting from the top left 2bit, would result in:
(001001**01** 111010**00** 110010**00**)
(001001**11** 11001000 11101000)
(11001001 00100111 11101001)

5-Experimental Results

An information hiding system, Secure Information Hiding System (SIHS) has been developed to provide confidentiality security service. SIHS employs an image file as a carrier to hide a message and focuses on Least Significant Bit (LSB) as one of the steganography techniques ***[8].*** The stego file does not reveal any difference in attributes like size, content etc from that of the original file. Hence it is difficult for someone to find out that this image contains a message.This is because the amplitude of the change is small, and therefore modulating the LSB does not result in a human- perceptible difference show Fig.(7).

In the experiment, we found that the size of information to he hidden relatively depends on the size of the cover-image (background). The message size must be smaller than segmentation background of the color image. With these increased levels of protection using encryption algorithm (DES). The call specified desirable criteria for such an algorithm :able to provide a high level of security, specified and easy to understand, publishable so that security does not depend on the secrecy of the algorithm, available to all users, adaptable for use in diverse applications, economical to implement in electronic devices, efficient to use ,able to be validated and exportable. The proposed system for steganography is more strong from attacks than any other existing system***.(show Table (4))***

Position Of pixe1	**Pixel of cover image** **R G B**	**Pixel of stego-image** **R G B**
350	**113,111,100**	**113,111,100**
400	**113,111,100**	**113,111,100**
530	**113,111,100**	**112,110,101**
10010	**255,255,255**	**225,255,254**
11110	**63,82,84**	**63,80,85**
12410	**77,73,71**	**77,75,68**
1046	**174,203,199**	**175,203,199**
1050	**255,255,255**	**255,255,255**
105111	**150,127,115**	**150,127,115**
115111	**150,127,115**	**150,124,115**
120111	**151,127,130**	**150,126,128**
14624	**255,255,255**	**255,255,255**
153200	**96,65,49**	**96,65,49**

The JSEG algorithm is tested on a variety of images. Fig.(6) and Fig.(7) shows the results of the segmentation background on images. Segmented images are dimmed to show boundaries between background and foreground. It can be seen that the results are quite good. Due to the lack of ground truth, however, we are unable to perform any objective evaluation or comparison with other segmentation methods. The JSEG Algorithm has 3 parameters that need to be specified by the user. The first one is a threshold for the color quantization process. It determines the minimum distance between two quantized colors. The second one is the number of scales desired for the image .The last one is a threshold for region merging after embedded encrypted data. These parameters are necessary because of the varying image characteristics in different applications.

To measure the difference between the original cover and stego-image we use the Peak Signal to Noise Ration (PSNR), which expressed as the following equation and ***(Table (3)*)*:-[8]***

$$\boldsymbol{PSNR=10Log_{10}}\frac{255^2}{MSE} \quad \text{..............................}(11)$$

and Mean-Square Error (MSE) is defined as:-

$$\boldsymbol{MSE=}\left(\frac{1}{H*W}\right)\sum_{1}^{H}\sum_{1}^{W}\left(X_{ij}-X'_{ij}\right)^2 \quad \text{.........}(12)$$

where H, W are the size of the cover image (H=256,W= 256 in this paper), x_{ij} : is the image.

Table(3)
Results PSNR and MSE.

image	***MSE***	***PSNR***
Figure-7	**0.162**	**56.036**

Table (4): *Results of pixel before and after embedded data using LSB-1,2bit insertion*

6-Conclusion

In this paper we give an idea to enhance the security of system by combining the two techniques (***steganography and cryptography***) and a new approach for segment background of color image called JSEG, is presented. The segmentation consists of color quantization and spatial segmentation. Applying the criterion to local image windows results in *J*-images, which can be segmented using a multi scale region growing method. Results show that JSEG provides good segmentation background on a variety of color images. It can enhance confidentiality of information and provides a means of communicating privately. Here message is first encrypted and then embed in segmentation background of color image file with help of steganographic system. The system security is further enhance by using a boundaries regions to embed the message. There are infinite number of steganography applications for digital image including copyright protection, feature tagging, and secret communication.

References

[1] Niels.Provos and Peter Honeyman ,"Hide and Seek: An Introduction to Steganography", IEEE1540-7993, June 2003

[2] Alain C. Brainos II East Carolina University, "Study of steganography and The Art of Hiding of information", November 5,2008

[3] John Wiley and Sons, "Computer Security and Cryptography" by Copyright 2007, page 288-294

[4] Man Young Rhee "Internet Security Cryptographic Principles, Algorithms and Protocols", by. School of Electrical and Computer Engineering, Copyright 2006, page 64

[5] S. Belongie, et al., "Color- and texture-based image segmentation using EM and its application to content-based image retrieval", Proc. Of ICCV, 2004

[6] Yining Deng, B. S. Manjunath and HyundooShin" ColorImage Segment-ation", Department of Electrical and Computer Engineering University of California, Santa Barbara,2006

[7] N.F. Johnson, J. Suhil, " Exploring Steganography: Seeing the Unseen," Computing practices, 2006, http://www.jjtc.com/pub/r2026.pdf

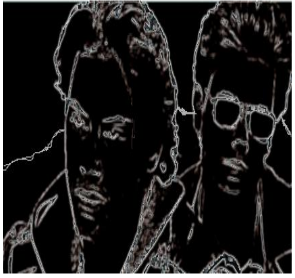
[8] Mr C.Raffert "Steganography & Steganalysis of images "Msc Comms Sys Theory 2005 http://www.stego.com

الخلاصة

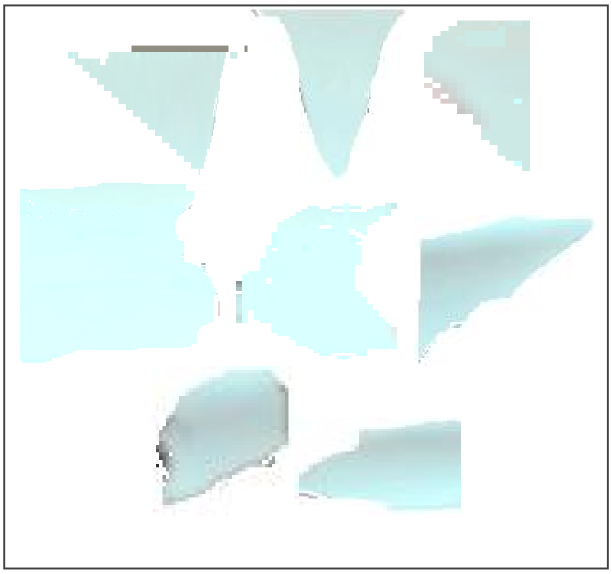
تم استحداث تقنية لاخفاء البيانات في خلفية المجزئة للصورة الملونة.تتضمن التقنية تحديد الخلفية الموجودة في الصورة الملونه بالاعتماد على القيم العليا والدنيا لحدود ومراكز المناطق المجزئة الموجودة في الصورة الملونة ومن ثم يتم اخفاء البيانات المشفره باحدى طرق التشفير في خلفية الصورة الملونة باستخدام طريقة البت الاقل لهاومن ثم مفارنتها مع الصورة الاصلية باستخدام معيار الربع الاصغر لمعرفة مقدار الاختلاف بين الصورتين.

a- original image(256×256).

b-result of color quantization with 13 color.



c-   J-image at scale 3.



e- Result of segment background of image using JESG method and then embedding data

e- Result of background of image using JESG

f- stego- image after embedding data and region merged

Fig.(7) Result of the concealment data in image.