ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

Enhanced Steganographic Method using 3-bit Grayscale Arithmetic Texture Synthesizer

K.R.Lavanya

Lecturer, Dept. of C.S.E., JNTUACEK, JNTU Anantapur, Andhra Pradesh, India

ABSTRACT: I propose an enhanced steganographic method that adopts a grayscale arithmetic texture synthesizer based on MRF for effective communication by increasing the confidentiality. Every synthesized cover image has eight gray values (30, 60, 90, 120, 150, 180, 210 and 240). The secret message is encrypted using AES algorithm before embedding in the cover image. Then the encrypted message is partitioned into bit sequence of 3-bit words (000 to 111) where each 3-bit word maps to eight gray values respectively. Left circular shift is performed on those gray values to embed different 3-bit words into different pixel values to enhance the security.

KEYWORDS: Steganography, Markov Random Field(MRF), Grayscale, AES algorithm, Left circular shift

I. Introduction

Now a day, internet communication forces us to protect data with high security to avoid hacking. We have three major ways to provide security: Cryptography, Watermarking and now steganography.

Steganography is a process of covering the secret message inside other file. As no one knows that a message is being sent through image, except the intended user, steganography is more secure than cryptography. Many steganographic methods have been projected for effective communication.

As the images traverse through internet, its size affects the length of the secret message that it cover. MRF[1,2,3] gray scale texture synthesizer is used here to overcome this drawback.

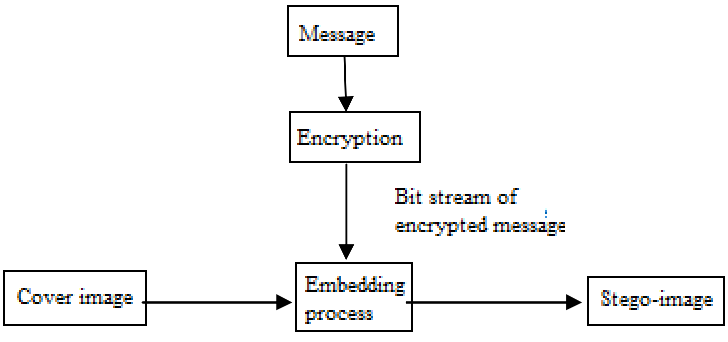

Fig. 1: Data Embedding in Cover image

II. Related work

1. *TEXTURE SYNTHESIS BASED ON MRF:*

This synthesis shows a dependency relationship among pixels with its neighbors. Based on MRF synthesizer, energy function E(x) can be computed to find the required size of a cover image. Required notations to define E(x) are listed below.

S= {0,1, …, G-1}, a set of gray values. For example, here G=8.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

x : Initial M x N texture in matrix form and x(i,j)ϵ S.

Ω= {x | x_t= x(i,j)ϵ S} ∀ 0 ≤ i ≤M and 0 ≤ j ≤ N.

θ_k: Weight of direction k.

k= min {1, P(Y)/P(X)}

The improved probability for all xϵ Ω is P(x):

$$P(x) = \frac{e^{-E(x)}}{\sum_{y\epsilon\Omega} e^{-E(y)}}$$

Where $E(x)= \sum_{i=1}^{M.N} W(x_i) + \sum_{i=1}^{M.N} \sum_{k=-n}^{n} H(x_i, x_{i+k})$

W(x_i) : the weight of pixel x_i and *n* is the size of neighborhood, let *n*=2, for 2^{nd} order neighborhood.
H(x,y)=H(y,x) and

$H(x_i,x_{i+k})= \theta_k \times I(x_i,x_{i+k})$

If $x_i=x_{i+k}$, $I(x_i,x_{i+k})$ = -1 else 1.

Algorithm:
Step 1: for j=1 to M.N, assign x_j= v, where v ϵ S.
Step 2: for j=1 to M.N,
  2.1: consider $y_i=x_i \forall\ i \neq j$ and assign y_i=v where v is randomly selected from S.
  2.2: k= min {1, P(Y)/P(X)}
  2.3: assign x ← y with probability k.
Step 3: Repeat 2^{nd} step until we reach convergence of the cover image.

III. PROPOSED ALGORITHM

Embedding Algorithm:
Input: α: number of dependent locations,
  C: cipher text, which is an encrypted secret message using AES algorithm and streamed as 3-bit words,
  k: constant value to signify the embedded pixels.
  T: number of left circular shifts to be applied on gray values with respect to 3-bit words in Table 1,
Step 1:
Let the starting embedding location as (0,0) and then scan the cover image in lexicographic order.
Step 2:
Every time read a 3-bit word x from the stream of cipher text C.
Step 3:
Apply left circular shift on gray scale values for T times and find the first pixel that come across while scanning, whose pixel value is x.
Step 4:
Increase its pixel value to x+k.
Step 5:
Find next α – dependent pixel locations by using the current pixel location.
Step 6:
Repeat Step 2 to Step 5 until entire cipher text is embedded.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

Description of Proposed algorithm:

We take eight grayscale values(30,60,90,120,150,180,210,240) to map with 3-bit words (000,001,010,011,100,101,110,111) respectively as shown in Table 1.

3-bit word	000	001	010	011	100	101	110	111
Pixel value	30	60	90	120	150	180	210	240

Table -1: Gray scale values with corresponding 3-bit words

For every 3-bit word, before embedding, apply left circular shift for T times to the gray scale values so that the different 3-bit words will be embedded into different pixels. Then select α fixed locations randomly to generate α dependent pixel locations based on last embedded pixel location.

Last embedded Pixel location	+	α locations	=	α dependent pixel locations

For example consider (a,b) be the last embedded pixel location and (x_1,y_1), (x_2,y_2) are the randomly selected fixed α locations, then

$((a+x_1) \bmod M, (b+y_1) \bmod N)$ and

$((a+x_2) \bmod M, (b+y_2) \bmod N)$ will be the dependent pixel locations

We use these pixels' values to calculate T by using the formula as follows:

$$T = \left(\sum_{i=1}^{\alpha} \left\lfloor {(p_i - 30)}/{30} \right\rfloor \right) \bmod 8 \quad \_\_\_\_ \text{eq(1)}$$

where p_i is the value of dependent pixel.

After calculating the T value, perform left shift for T times and find the first pixel with the corresponding gray scale value and increase its pixel value by k units.

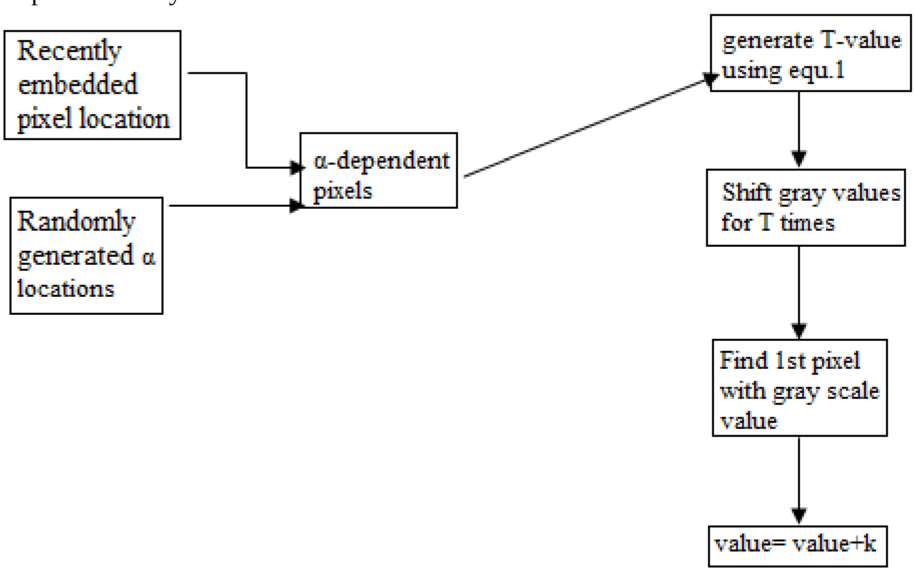

Fig.2: Process of embedding

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

Extraction Algorithm:

Input: α: number of dependent locations,
Stego-image of size MxN,
k: constant value that signified the embedded pixels.
T: number of left circular shifts to be applied on gray values with respect to 3-bit words in Table 1,

Step 1:
Set the pixel (0,0) as the latest embedded and scan the stegoimage in the order of embedded process.

Step 2:
Find all the pixels' locations with value v∈{30+k, 60+k, 90+k, 120+k, 150+k, 180+k, 210+k, 240+k}, decrease pixel value of those pixels by k.

Step 3:
Apply left circular shift for the gray scale values (30, 60, 90, 120, 150, 180, 210, 240) for T times with respect to the 3-bit words (000,001,010,011,100,101,110,111) according to Table-1.

Step 4:
Compute α dependent locations with respect to the latest embedded location.

Step 5:
Calculate T by eq.(1)

Step 6:
Repeat step 2 to step 5 until no pixel with the pixel value v ∈{30+k, 60+k, 90+k, 120+k, 150+k, 180+k, 210+k, 240+k}.

IV. SIMULATION

Consider the following assumptions to simulate an example of embedding an encrypted message.
1: An stegoimage of size – 4x8 [0:3, 0:7] and start at left most pixel (0,0) as latest embedded pixel at first.
2: let C: Cipher text with the bit sequence of {101,010,011,001}
3: α = 2 and randomly selected fixed pixel locations (α) are : {(1,1),(2,3)}and k=4
4: let T=0 initially, number of left circular shifts to be performed on gray values with respect to 3-bit words, as shown in Table -1

EMBEDDING PROCESS:

1. To embed the first word 101, find the first pixel with the gray value 180, as T= 0.
   From the fig.(a), pixel (0,7) contains 180 which corresponds to 101. So update (0,7) value from 180 to 180+4=184 as shown in
   .(b).
   As (0,7) is latest embedded pixel, α dependent locations are:
   (0,7) +(1,1) = (1,0)
   (0,7)+(2,3)= (2,2). These are the α locations with pixel values 30, 150 respectively.
   then T= $\left\lfloor \frac{30-30}{30} \right\rfloor + \left\lfloor \frac{150-30}{30} \right\rfloor$ = 0+ 4 = 4 mod 8=4
2. So apply left circular shift T =4 times on Table -1, to embed next pixel 010. After performing shift operation, 210 is the gray scale value w. r. t. 3-bit word 010 now. From the fig.(a), pixel (1,1) contains gray scale value 210. Increase its pixel value to 210+4= 214, shown in fig.(b).

   As(1,1) is the latest embedded pixel, α dependent pixel locations are:
   (1,1)+(1,1)=(2,2) and
   (1,1)+(2,3)=(3,4). These are the new α dependent locations with gray values 150, 60 respectively.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

Then T= $\left\lfloor\frac{150-30}{30}\right\rfloor + \left\lfloor\frac{60-30}{30}\right\rfloor$ = 4 + 1 = 5 mod 8 =4

3. Apply left circular shift for 5 times, to embed 3rd 3-bit word 011.. After performing shift operation, 150 is the gray scale value w. r. t. 3-bit word 011 now. From fig.(a), pixel (1,2) contains gray scale value 150. Increase its pixel value to 150+4=154, shown in fig.(b).

   As (1,2) is the latest embedded pixel, α dependent pixel locations are:
   (1,2)+(1,1)=(2,3) and
   (1,2)+(2,3)=(3,5). These are the new α dependent locations with gray values 210, 90 respectively.

   Then T= $\left\lfloor\frac{210-30}{30}\right\rfloor + \left\lfloor\frac{90-30}{30}\right\rfloor$ = 6 + 2 = 8 mod 8 =0

4. So no need to apply left circular shift to embed 4th 3-bit word 001.
   From fig.(a), pixel (3,5) contains the gray value 90 which corresponds to 001. Increase it to 90+4=94, shown in fig.(b).

90	30	30	120	150	150	60	180
30	210	150	120	120	180	180	210
30	120	150	210	60	240	240	180
210	210	150	60	30	90	90	210

Fig.(a) pixel values of 4x8 cover image

90	30	30	120	150	150	60	184
30	214	154	120	120	180	180	210
30	120	150	210	60	240	240	180
210	210	150	60	30	94	90	210

Fig.(b) pixel values of 4x8 stego-image image

V. CONCLUSION AND FUTURE WORK

The proposed method use 3-bit gray scale values to increase the payload capacity of the cover image. So huge message can be embedded in the given cover image. As any one needs α fixed locations to extract the cipher text, it's hard to retrieve the secret message without knowing α locations, which are sent to the receiver using secret key. So this proposed method helps to overcome the drawback of limited size of cover image for embedding.

REFERENCES

1. G.R. Cross and A.K. Jain, 'Markov Random Field Texture Models', IEEE transactions on Pattern Analysis and Machine Intelligence, vol.5, No. 1, 25-39, 1983.
2. F.J.Chang, 'A Steganographic Method Using MRF-Synthesized Textures as Cover Images', M.S. Thesis, National Tsing Hua University, Hsinchu, Taiwan, April 2011.
3. C.L. Liu and S.R. Liao, 'High-performance JPEG steganography using complementary embedding strategy', Pattern Recognition, Vol. 41, 2945-2955, 2008.

4. C.C. Chen and C.C.Chen, 'Texture Synthesis: A Review and Experiments', Journal of Information Science and Engineering, Vol. 19, No. 2, 371-380, 2003.
5. N.F. Johnson and S. Jajodia, 'Steganalysis of images Created Using Current Steganographic Software', Int'l Workshop in Information Hiding, Berlin, Vol. 1525, 273-289, 1998.
6. Kuo-Chen Wu and Chung-Ming Wang, 'Steganography Using Reversible Texture Synthesis', IEEE Transactions on Image Processing, Vol. 24, No. 1, 130-139, January 2015.
7. J. Fridrich, M. Goljan and R. Du, 'Detecting LSB steganography in color and gray-scale images', IEEE MultiMedia, Vol. 8, No. 4, 22-28, Oct/Dec 2001.
8. C. Han, E. Risser, R. Ramamoorthi, and E. Grinspun, 'Multiscale texture synthesis', ACM Trans. Graph., vol. 27, No. 3, 2008.
9. Z. Ni, Y.-Q. Shi, Ansari, and W. Su, 'Reversible data hiding', IEEE Trans. Circuits Syst. Video Technol., Vol. 16, No. 3, 354-364, Mar.2006.
10. A. A. Efros and W. T. Freeman, 'Image quilting for texture synthesis and transfer', in Proc. 28th Annu. Conf. Comput. Graph. Interact. Techn., 2001, 341-346.

BIOGRAPHY

K. R. Lavanya is a Lecturer in the Computer Science and Engineering Department, JNTUACE, Kalikiri. She received Master of Technology (M. Tech) degree in 2012 from JNTUK, Andhra Pradesh,India. Her research interests are Information Security, Image Processing, etc.