A Covert Communication Model-Based on Image Steganography

Mamta Juneja, UIET, Panjab University, Chandigarh, India

ABSTRACT

With the more and more advancement in technology, internet has become the most important medium for all kinds of confidential as well as non-confidential communications. Security is the major issue for such communications and steganography is most widely accepted tool for information security. An effort has been made in the present paper to propose a secured model for communication using image steganography. It presents two components based LSB steganography method, adaptive LSB based steganography method for embedding data in high and low transition parts of an image respectively. Hybrid edge detection filter is proposed to divide an image in low and high transition areas. AES (Advanced Encryption Standard) and Randomization is incorporated to provide two-tier security. Comparison analysis of output results with other existing techniques on basis of capacity, imperceptibility is giving the proposed approach an edge over others. The proposed approach has been thoroughly tested for various steganalysis attacks like visual analysis, histogram analysis, chi-square, and RS analysis and could sustain all these attacks very well.

Keywords: *Adaptive LSB, AES, Hybrid Feature Detection, Random Pixel Embedding, Steganography, Two Component Based LSB*

1. INTRODUCTION

Markus Kahn (1995) defines Steganography as an art and science of communicating in a way which hides the existence of the communication. In contrast to cryptography, where the enemy is allowed to detect, intercept and modify messages without being able to violate certain security premises guaranteed by a cryptosystem, the goal of steganography is to hide messages inside other innocent messages in a way that does not allow any enemy to even detect that there is a second message present.

Initial work on LSB steganography was on LSB Substitution and was explored by Chang et al. (2002), THIEN et al. (2003), Wang et al. (2000, 2001), Chang et al. (2003, 2006), Chan et al. (2001, 2004), which substitutes the same number of bits of each and every pixel of input host image for hiding the secret text or message

DOI: 10.4018/ijisp.2014010102

so give rise to pair of values (PoVs) and are easily attacked by Chi-square Test given by West fled et al. (1999), Provos et al. (2002), Stanley (2005). LSB Matching introduced by Ker et al. (2004) and researched by Mielikainen (2006), LI (2009), Luo et al. (2010), and Kumar et al. (2012) was attacked by Ker (2005) based on the Center of Mass (COM) of the Histogram Characteristic Function (HCF).Adaptive LSB was worked on by Lie et al. (2000), Liu et al. (2004), and Kekre et al. (2008) is based on variable number bits substitution but are unable to utilize HVS masking characteristics completely and are affected by edge masking effect. PVD methods are among the most popular method which was explored by Wu and Tsai (2003), Park et al. (2005), Wu et al (2005), Yang et al (2006), Jung et al. (2008), Liu et al. (2008), Wang et al. (2008), Yang et al. (2008), Maleki et al (2011), Liao et al. (2011), and Mandal et al. (2011). It follows the principle that the edge areas being high in contrast, intensity, transitions can tolerate more changes than smooth areas .But techniques based on these are unable to mark the difference in edge features and texture features so embed data in both. Moreover were complex to work and were easily attacked by Zhang et al. (2004). Edge detection Filter based technique was utilized by Alwan et al. (2005), Negi et al. (2006), Hempstalk (2006), Singh et al. (2007), Chen et al. (2010), Hussain (2011), and Bassil et al. (2011) for steganography in Gray images. But the advancement in image technology to RGB leads to steganography application for color images. Pixel indicator techniques introduced by Gutub et al. (2008), 2009), and Gandharba et al. (2011) for color images had a major drawback of treating all color components (red, green, blue) equally contradicting Hecht principle, which reveals that the visual perception of intensely red objects is highest and then of intensely Green objects and is least for intensely blue objects i.e. red plays the most significant and Blue plays a least significant role in color formulation. So, we can integrate maximum changes in Blue component and average changes in green component and least change in red component without making much difference in color image. Color Component based Techniques researched by Imran et al. (2007), Chang et al. (2008), Roque et al. (2009), and Mandal et al. (2012) were not fully tested fully for all types of attacks like targeted and universal and were focused on single component. They didn't utilize cryptography and Random Sequence generator techniques to make these techniques better resistant to attacks. Chen et al. (2010) proposed steganography technique using hybrid filter but tested it for gray images moreover didn't test it for targeted and universal attacks. Capacity of 2.8 bpp (bits per pixel)is good but highest PSNR value attained is 28.6 which is very low. Mandal (2011) achieved 49% PSNR but didn't even mention capacity factor and Hussain et al. (2011) achieved highest PSNR for very small text messages. Liao et al. (2011) achieved 39% PSNR with good capacity but also explicitly mentioned this in their research paper that they targeted quality and capacity from tradeoff between capacity, quality and robustness.

They compromised resistance to attacks for quality and capacity. Kekre et al. (2009), and Husain et al. (2010) also worked on quality and capacity but successfully attacked by (Singh et al., 2012).The three most required evaluation criteria's for any good steganography techniques are Robustness, Imperceptibility and Capacity. But there is no technique so far for color images which would target all these criteria's fully. So there is urgent requirement of technique which would provide good capacity and high PSNR value and resistant to all targeted as well as universal steganalysis attacks for color images. The Proposed research aims to develop an improved steganography approach for color images with higher imperceptibility/quality, large capacity/payload and better in robustness/resistance to attacks. It will incorporate cryptography to achieve high security and random pixel embedding to attain high immunity to attacks. It would be highly immune to any environmental disturbances like noise due to hybrid filtering.

2. DESIGN PHASE OF PROPOSED MODEL

The proposed model provides improved LSB based steganography technique, which will work in a spatial domain, for hiding data in a 24-bit bitmap color image. It integrates the following three new techniques:

1. Hybrid feature (line /edge /boundary / shape) detection technique combining Canny and Hough transform for bifurcating an image into edge and smooth areas;
2. Two Component based Least significant bit (LSB) Substitution Technique for hiding encrypted messages in edges of images;
3. An Adaptive LSB substitution technique for hiding messages to smooth areas.

It is integrating the following for improved steganography:1) Hybrid feature (line /edge / boundary /shape) detection technique combining Canny and Hough transform for bifurcating an image into edge and smooth areas 2)Two Component based Least significant bit (LSB) Substitution Technique for hiding encrypted messages in edges of images 3)Adaptive LSB substitution technique for hiding messages to smooth areas.

In addition to above, Advanced Encryption Standard (AES) is used to encrypt hidden message text/file to provide better security by combining Steganography with Cryptography. With this combination even if message would be detected would not be understood by intruder due to encryption. Random pixel embedding technique is also been incorporated to hide data at random pixels in cover image so that couldn't be attacked by sequential attacks. This provides the two tier security to hidden data.

The proposed research is the direct implementation of the principle that edge areas being high in contrast, color, density and frequency can tolerate more changes in their pixel values than smooth areas, so can be embedded with a large number of secret data while achieving high quality of the stego-image. The various algorithms utilized in proposed system are as follows.

2.1. Hybrid Edge Detection Technique for Extracting High and Low Transition Areas

A new hybrid edge detection technique for extracting high transition (edges) and Low transition (smooth) areas from an image is being proposed. It integrates the canny edge detection proposed by Canny (1986) and the enhanced hough transform edge linking technique given by Hough (1962).

2.1.1. Detection of High Transition Areas (Edges) Using Canny

The Canny edge detector is widely considered to be the standard edge detection algorithm in the industry. It is known as optimal edge detector due its good detection, good Localization and minimal false edges .It uses a multi-stage algorithm to detect a wide range of edges in images.

Algorithm:

1. **Smoothing:** Blurring of the image to remove noise;
2. **Finding gradients:** The edges should be marked where the gradients of the image has large magnitudes;
3. **Non-maximum suppression:** Only local maxima should be marked as edges;
4. **Double thresholding:** Potential edges are determined by thresholding;
5. **Edge tracking by hysteresis:** Final edges are determined by suppressing all edges that are not connected to a very certain (strong) edge.

2.1.2. Edge Linking by Hough

Edge Linking is implemented using global method named Hough Transform. Classical Hough Transform can locate regular curves like straight lines, circles, parabolas, ellipses,

etc. Generalized Hough Transform can be used where a simple analytic description of feature is not possible. This research proposes an enhanced Hough transform by combining both classical and generalized Hough transform to extract lines, edge boundaries, circles and shapes.

Algorithm for Enhanced Hough Transform:

1. **Algorithm for Line detection:** Edge detection is often used as preprocessing to Hough transform. The input image must be a thresholded edge image. The magnitude results computed by the canny operator can be thresholded and used as input. For an input image I of M × N:
   a. Locate the HT coordinate system;
   b. Identify the ranges for θ and ρ. Let the range for for θ be between θl and θh, and the range for ρ between ρl and ρh;
   c. Choose quantization intervals δθ and δρ for θ and ρ respectively;
   d. Discretize the parameter space of ρ and θ using sampling steps δρ and δθ. Let θd and ρd be 1D arrays containing the discretized θ and ρ;
   e. Let A(T, R) be an array of integer counter; initialize all elements of A to zero, where R =ρh-ρl/δρ and and T = θh-θl/δθ;
   f. Let L(T,R) be an array of a list of 2D coordinates;
   g. For each image pixel (c, r), if its gradient magnitude g(c, r) > τ, where τ is a gradient magnitude threshold:

for i=1 to T
ρ = c * cos(θd(i)) + r * sin(θd(i))
find the index k, for the element of ρd closest to ρ
increment A(i,k) by one
add point (c,r) to L(i,k)

   h. Find all local A(ip, kp) such that A(ip, kp) > t, where t is a threshold;
   i. The output is a set of pairs (θd(ip), ρd(kp)) and lists of points in L(ip, kp), describing the lines detected in the image, along with points located on these lines.
2. **Algorithm for Edge linking:**
   a. Obtain a thresholded edge image;
   b. Specify subdivisions in the ρθ-plane;
   c. Examine the counts of the accumulator cells for high pixel concentrations;
   d. Examine the relationship (principally for continuity) between pixels in a chosen cell.

Continuity here normally means distance between disconnected pixels and a gap in the line can be bridged if the length of the gap is less than a certain threshold.

Advantages of New Hybrid Edge Detector:

1. It is using probability for finding error rate;
2. It is easier to locate various features like lines, edges, triangle, circles and boundaries with same accuracy and efficiency with this method;
3. It helped to improve signal to noise ratio which was one of the main objective of this research;
4. It is resistant to all kinds of noise, disturbances and provides good detection;
5. It is tolerant towards gaps in the boundary line and occlusion in the image;
6. It is robust to partial deformation in shape and can detect multiple occurrences of a shape in the same pass.

2.2. Adaptive LSB Substitution for Smooth Areas

We have used the algorithm provided by Kekre (2009) and modified it to get smoother areas. In this approach a variable number of LSBs would be utilized for embedding secret message bits, in accordance with this algorithm.

For all pixels across smooth areas:

1. If the value of the red component (sp_rc), green component (sp_gc), and blue component (sp_bc) of the smooth pixel spi is in the range of 240 to 255 (i.e.,

240<=spi_rc<=255, 240<=spi_gc<=255, and 240<=spi_bc<=255) then utilize all of the bits of the blue component for embedding data. This is done by first checking that the 4 MSBs all equal 1;
2. If the value sp_rc, sp_gc, sp_bc is in the range of 224 to 239 (i.e., 224<=spi_rc<=239, 224<=spi_gc<=239 and 224<=spi_bc<=239) then utilize 6 LSBs of the blue component for embedding of data This is done by checking that the first 3 MSBs all equal 1;
3. If the value sp_rc, sp_gc, sp_bc is in the range of 192 to 223 (i.e., 192<=spi_rc<=223, 192<=spi_gc<=223 and 192<=spi_bc<=223) then utilize 5 LSBs of the blue component for embedding of data. This is done by checking that the first 2 MSBs both equal 1;
4. If the value sp_rc, sp_gc, sp_bc is in the range of 0 to 191 (i.e., 0<=spi_rc<=191, 1<=spi_gc<=191 and 0<=spi_bc<=191) then utilize 4 LSBs of the blue component for embedding for embedding of data. This is done by checking that the 1st MSB equals 1.

This is illustrated in Table 1.

For all of the components (red, green, blue) of each and every pixel in a color image across the smooth areas (except those already embedded by the above algorithm), the following embedding process is employed:

1. If the value of the current pixel component (the first 4 MSB's are all 1's) say for example, the cpci, is in the range of 240 <= cpci>=255, then we utilize 4 LSBs of that component for embedding;
2. If the value of the cpci (First 3 MSB's are all 1's), is in the range of 224 <= cpci>=239 then we utilize 3 LSBs of that component for embedding;
3. If the value of the cpci (the first 2 MSB's are all 1's), is in the range of 192 <= cpci>=223 then we embed 2-bits of secret data into the 2 LSB's of that component;
4. And in all other cases where values are in the range of 0 <= cpci>=192 we embed 1-bit of secret data into 1 LSB of that component.

This is illustrated in Table 2.

A similar procedure is adapted for extracting the hidden text from the image.

2.3. Two Components-Based LSB Substitution for Edge Areas

In this method, the 8-bits of the first component (blue component) for the image pixels would be

Table 1. Per pixel embedding chart

sp_rc	sp_gc	sp_bc	Utilized Bits	Total Bits Available/ Pixel
240-255	240-255	240-255	sp_rc -4 LSBS sp_gc-4LSB**s** **sp_bc-8LSBs**	16 bits
224-239	224-239	224-239	sp_rc -3 LSBS sp_gc-3 LSB**s** **sp_bc-7 LSBs**	13 bits
192-223	192-223	192-223	sp_rc -2 LSBS sp_gc-2 LSB**s** **sp_bc-6 LSBs**	10 bits
0-191	0-191	0-191	sp_rc -1 LSB sp_gc-1 LSB **sp_bc-5 LSBs**	7 bits

Table 2. Per pixel component embedding chart

Range of the Smooth Pixel Components (sp_rc or sp_gc or sp_bc)	Utilized Bits/Component
240-255	4
224-239	3
192-223	2
0-191	1

replaced with secret text message bits followed by a secret message being embedded into the 4 least significant bits of the green component.

For all edge pixels retrieved through hybrid feature detection:

1. Read each edge pixel provided by the hybrid feature detection that was described in Section 3.1. Each edge pixel can be represented as:

epi={ Ri[r0, r1, r2, r3, r4, r5, r6, r7], Gi [g0, g1, g2, g3, g4, g5, g6, g7], Bi [b0, b1, b2, b3, b4, b5, b6, b7]}

Here, I is a 24-bit image so each of its pixels is made up of 3 color components red(R), green(G), and blue (B). Each of which is 8-bits in length. i is the index of the ith pixel, ri is the ith bit of the color component R, gi is the ith bit of color component G, and bi is the ith bit of the color component B:

2. Embed message bits in epi (Gi [g4 ,g5, g6, g7], Bi [b0, b1, b2, b3, b4, b5, b6, b7]) in the sequence b7 to b0 and then g7 to g4;
3. Embed data until:
   a. The end of the message data is not reached -OR-;
   b. The end of the edge pixels is reached;
4. If the end of the message data is reached then the embedding is over and in the other case embed the rest of the message data using by using adaptive LSB approach given in section P2.

The utilizations bits in this technique are 12-bits out of a total of 24-bits of pixels.

A similar procedure is used for extracting message bits from edge pixels.

2.4. Encryption Using Advanced Encryption Standard (AES)

An input text file is encrypted using the standard encryption technique AES, as defined in FIPS (2001), in order to provide two tier security for the proposed system. It ensures that the message will not be understood by any, even in the case that its existence is disclosed due to being encrypted.

Algorithm: The algorithm begins with an Add round key stage followed by 9 rounds of four stages and a tenth round of three stages. This applies for both encryption and decryption with the exception that each stage of a round the decryption algorithm is the inverse of its counterpart in the encryption algorithm. The four stages are as follows:

1. Substitute bytes: It carries byte-by-byte substitution during the forward process;
2. Shift rows: It shifts the rows of the state array during the forward process;
3. Mix columns: It mixes up the bytes in each column separately during the forward process;
4. Add round key: It adds the round key to the output of the previous step during the forward process.

The tenth round simply leaves out the Mix Columns stage. The first nine rounds of the decryption algorithm consist of the following:

1. Inverse Shift rows;
2. Inverse Substitute bytes;
3. Inverse Add Round Key;
4. Inverse Mix Columns.

Again, the tenth round simply leaves out the Inverse Mix Columns stage.

2.5. Random Pixel Embedding

Random pixel embedding, as given by Schneier et al. (2003), has been implemented in the present work by using the Linear Congruential Generator (LCG). It is also known as the Pseudo Number Generator (PRNG), which is probably one of the most commonly used techniques out of the family of pseudorandom techniques, and is used for generating random numbers. With a given seed it will keep generating random numbers, this is why it selects random pixels to embed in both edge and smooth areas:

1. **Random Pixel Embedding:** Before being embedded into any pixel, a random pixel is selected using the function random_pixel_generator (), which is defined below;
   a. Initialize the values for the increment, multiplier, initial seed, and the maximum_ possible_value. (An initial seed is any value from 0 to 16,777,215 for a 24-bit color bitmap image and the maximum possible value is 1,677,215);
   b. Assign values for the seed, increment, multiplier, and the maximum_possible_value as long as the 4 conditions listed below are satisfied:
      i. The increment must be relatively prime to maximum possible value;
      ii. multiplier-1 must be a multiple of every prime p that divides the maximum_possible_value;
      iii. multiplier-1 must be a multiple of (a number) if the maximum_possible_value is a multiple of a number;
      iv. The seed, increment, multiplier, and maximum_possible_value must all be greater than 0;
   c. Compute the Random_Pixel_Value= Multiplier * Seed + Increment mod (maximum possible value);
   d. Repeat step c while keeping the values of the increment, multiplier, and maximum possible value as the same. However, the random_pixel_value generated becomes the seed for the next random number, in order to generate all values from 1 to the maximum possible value;
   e. Return the Random_Pixel_Value.
      i. Do this so that the output will be a random pixel number;
2. **Random Pixel Extraction:** During extraction the same procedure is repeated while facilitating the selection of the same pixels. Random pixel extraction will select the same set of pixels for the same set of value increments, multipliers, initial seeds, and the maximum range;

3. DEVELOPMENT PHASE OF PROPOSED MODEL

The proposed model comprises of two components:

1. Embedding Module;
2. Extracting Module.

3.1. Embedding Module

Embedding is the process of hiding the embedded message generating the stego image. Hiding information may require a Stego key which is additional secret information, such as password, required for embedding the information .For example, when a secret message is hidden

within a cover image, the resulting product is stego image (stego object).

The main algorithm for the Embedded stage can be listed as follow:

1. Input the secret text/image file that to be hidden in the cover image;
2. Select the cover image (BITMAP file) from list of stored Image files and the text files;
3. Extraction of Input cover image to Edge and smooth areas using new hybrid feature detection filter described in section 2.1 and Procedure 3.3;
4. Calculate the size of the secret text;
5. Secret text data is first encrypted using the standard Advanced Encryption standard described in section 2.4 and Procedure 3.4;
6. Substitute the encrypted secret characters from step 5 to cover image obtained from step 3 randomly explained in section 2.5 and Procedure 3.5:
    a. For edge areas, embed secret text data using New Two Component based LSB Substitution Technique described in Section 2.3 and Procedure 3.1.1;
    b. For smooth areas, embed secret text data using Adaptive LSB method section 2.2 described in Procedure 3.1.2.

3.1.1. Procedure for Embedding for Edge Areas

1. Extract all the edge pixels in the given image and store it in the array called Pixel-Array;
2. Extract all the characters in the given text file and store it in the array called Character- Array;
3. Extract all the characters from the stego key in key array;
4. Choose first edge pixel and pick characters from Key-Array and place it in 8 bits of first component of pixel. If there are more characters in Key- Array, then place rest in the 4 bits of its second component and then to next pixel and so on till there are characters in key-array;
5. Place some terminating symbol to indicate end of the key. '0' has been used as a terminating symbol in this algorithm;
6. Place characters of Character- Array in each 8 bits of first component (blue) and 4 bits of second component (green) of next pixels by replacing it and so on till all the characters has been embedded. Again place some terminating symbol to indicate end of data;
7. Obtained Stego image will hide all input Characters.

3.1.2. Procedure for Embedding for Smooth Areas

For All RED-GREEN-BLUE components of all pixels:

1. Calculate no. of bits available for embedding as given in section 2.2;
2. Place the remaining characters from character into available bits of pixel array.

Repeat steps (1; 2) till we reach end of character array.

3.2. Extracting Module

Extracting is the process of getting the embedded message from the stego image.

The main algorithm for the embedded stage is as follow:

1. Extraction of Input cover image to Edge and smooth areas using new feature detection filter as in section 2.1 and Procedure 3.3;
2. Extraction of secret text message from stego image is carried from random pixels of cover image using Function defined in section 2.5 and Procedure 3.5;
3. For edge areas: Extract data from 8 bits of BLUE component and 4 least significant bits of GREEN component using section 2. 3 and Procedure 3.2.1;

4. For smooth areas: Extract data from adaptive no. of bits using Section 2.2 and Procedure 3.2.2;
5. Apply AES Decryption method described in section 2.4 and Procedure 3.4.

3.2.1. Procedure for Extraction for Edge Areas

Extract all the pixels in the given image and store it in the array called Pixel-Array. Now, start scanning pixels from Pixel-Array and keep extracting key characters from first and second (partial) components of all pixels to Key-Array till we get the terminating symbol. If this extracted key matches with the key entered by the receiver, then again start scanning next pixels and extract secret message characters from first (blue) and second (partial green) component of next pixels and place it in Character Array till we get terminating symbol.

3.2.2. Procedure for Extraction for Smooth Areas

For All Red-Green-Blue Components of All Pixels:

1. Calculate no. of bits available for embedding as given in section 2.2;
2. Now, start scanning pixels from Pixel-Array and keep extracting characters from the no. of bits determined by step a in character array till we get terminating symbol.

3.3. Procedure for Feature Detection

1. The algorithm reads a 24-bit color image denoted by CI= {cp1, cp2, cp3,…..cpn-1} where cpi is the ith pixel in the image and n is the total number of pixels. In the same context, every color pixel cpi can be represented as cpi = {rci[rc0,….rc7], gci [gc0,…..gc7], bci [bc0,……bc7]} where i is the index of the ith pixel, rci is the ith bit of color component rc, gci is the ith bit of color component gc, and bci is the ith bit of color component bc. Here CI is a 24-bit image so each of its pixels is made up of three color components each of which is of length 8 bits;
2. The algorithm converts CI into a Gray image denoted by f(CI)=GI where GI is gray image. The purpose of this conversion is to ease the processing of subsequent steps;
3. Three parameters 1) the size of the Gaussian filter, 2) a low threshold, and 3) a high threshold are automatically chosen where results of filter are optimal;
4. The Canny edge detection algorithm is executed on GI as described in section 2.1.1 using the three parameters selected in step 3. The results are a collection of lines, curves, and points denoting the edges or the boundaries of the objects in the image GI. The pixels that constitute the extracted edges are represented as EP= {ep1, ep2, ep3 …epr-1} where ej is the jth pixel that makes up the edges and r is the total number of these pixels.

Thereafter we apply Enhanced Hough transform to extract various other features like shapes lines and circles.

The output image after applying Canny edge detector is having distorted edges which are not properly joined with each other so edge linking is required to fill those edge gaps. Therefore a global edge linking technique i.e. Hough Transform has been applied on the output image obtained by canny so as to get more refined edges. Hough Transform would even detect line, edges, linkings and shapes which were not traceable through Canny. The edge pixels retrieved after applying Hough Transform given in Section 2.1.2 are represented as HEP= {hep1, hep2, hep3 …hepq-1} where hepj is the jth pixel in the image and q is total number of edge pixels.

3.4. Procedure for Encryption Using AES

Pseudo code of AES round transformation:

```
Round (State, ExpandedKey[i])
{
SubBytes (State);
ShiftRows (State);
MixColumns (State);
AddRoundKey (State, ExpandedKey[i]);
}
FinalRound (State, ExpandedKey [Nr])
{
SubBytes (State);
ShiftRows (State);
AddRoundKey (State, ExpandedKey [Nr]);
}
```

3.5. Procedure for Random Pixel Embedding

```
INPUT: (Key, Seed)
OUTPUT: random_data, (Key', Seed')
random_data = F(Key, Seed)
Key' = F (Key, Seed+1)
Seed' = F (Key', Seed)
returnrandom_data
```

4. RESULTS AND PERFORMANCE ANALYSIS OF PROPOSED MODEL

4.1. Evaluation Criteria

Steganography techniques are broadly evaluated in three aspects:

Criteria I: Imperceptibility/Quality;
Criteria II: Capacity or Payload;
Criteria III: Robustness or Resistance to Attacks.

1. **Imperceptibility/Stego-Image Quality:** It is the scale to measure the quality of stego image after hiding the details inside. It provides the imperceptibility and invisibility measurement and is highest if the differences in cover and stego image are not visible. As we all know, the higher the stego-image quality, the more invisible the hidden message. Therefore, the stego-image quality is a very important criterion to use when we evaluate the performance of a steganographic technique. We can judge whether the stego-image quality is acceptable to the human eye by using Peak Signal-to-Noise Ratio (PSNR):

$$PSNR = 10LOG_{10}\frac{255^2}{MSE}dB$$

PSNR can also be used as metrics to measure the degree of imperceptibility:

$$MSE = \left(\frac{1}{MXN}\right)\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}(P(x,y)-P'(x,y))^2$$

where M and N are the number of rows and number of columns respectively of the cover image, P(x,y) is the pixel value from the cover image, P'(x,y) is the pixel value from the stego-image. Signal to noise ratio quantifies the imperceptibility, by regarding the message as the signal and the message as the noise:

2. **Payload/Hiding Capacity:** The payload indicates the maximum number of bits that can be hidden with an acceptable resultant stego-image quality. Because the scheme would be of no value if the stego-image turned out seriously distorted despite the fact that it can hold a large amount of secret data, the hiding capacity does have its limit, especially when it comes to the binary image. We can say that a scheme does have its contribution to this field of research if it proves to either increase the payload while maintaining an acceptable

stego-image quality or improve the stego-image quality while keeping the hiding capacity at the same level, or better if it can get both promoted;

3. **Robustness/Resistance to Attacks:** It is resistance of stego image to various steganalysis attacks. It is immunity of stego image to all types of manipulations, operations carried on it and successfully transferring the hidden information. Steganalysis is the science of detecting hidden information. The main objective of Steganalysis is to break steganography and the detection of stego image is the goal of steganalysis. Steganalysis algorithms introduce statistical differences between cover and stego image. So Robustness of system is defined as defeating such attacks and successfully transferring information without anyone even knowing its existence.

4.2. Results for Criteria I and II

The outcome results for PSNR and capacity are shown in Tables 3 and 4 on various cover images after embedding a confidential text file, while varying the size of the text file and cover image.

4.3. Comparison Analysis with Existing Techniques for Evaluation Criteria I and II

The comparison of existing techniques with proposed approach on the basis of PSNR and capacity is shown in Table 5.

From Tables 5-9, we can say that our proposed approach has proven better in Evaluation Criteria I and II than already exisiting techniques. It provides better imeperceptibilty/quality and hiding capacity than previous known techniques.

4.4. Evaluation Analysis for Criteria III: Robustness/ Resistance to Attacks

The robustness of the proposed technique was thoroughly tested through various steganlysis attacks. These included visual analysis and

Table 3. Embedded data, % of pixels used in the image, % of changed bytes, average number of bits per pixel, MSE, and RMSE for test images

	Embedded Data	% of Used Pixel in Image	% of Changed Bytes	Avg. # Bits/ Pixel	MSE	RMSE
LEENA	261,121	0.4545	42.65	1.47	0.055	0.235
BABOON	267,134	0.4552	39.54	1.13	1.447	1.2.33
PEPPER	267,135	0.4545	41.39	1.13	1.433	1.197
FAMILY	270,936	0.4545	41.73%	1.5	0.01	0.122

Table 4. Embedded data, mean, standard deviation, PSNR, capacity for images

	Embedded Data	Mean	Standard Deviation	PSNR	Capacity
LEENA	261,121	256.5	361.3	60.70	806,912
BABOON	267,134	91	129.4	46.52	106,496
PEPPER	267,135	92.5	129.4	46.57	102571
FAMILY	270,936	213	299.8	66.37	544,768

Table 5. Capacity and PSNR value comparison of existing techniques

Technique	OPAP(3Bits)		LSB(3Bits)		OLSB(3Bits)		ALSB, HSV	
Author	(Chan et al., 2004)		(Chan et al., 2004)		(Wang et al., 2001)		(Lie et al., 1999)	
	Capacity	PSNR	Capacity	PSNR	Capacity	PSNR	Capacity	PSNR
LEENA	786,432	40.7	786,432	37.92	786,432	40.7	786,432	37.92
BABOON	786,432	40.7	786,432	37.92	786,432	40.7	786,432	37.92
PEPPER	786,432	NA	786,432	NA	786,432	NA	786,432	NA

Table 6. Capacity and PSNR value comparison of existing techniques (cont'd.)

Technique	Side Match		PVD		Adaptive LSB		PVD Modulus	
Author	(Chang et al., 2004)		(Wu et al., 2003)		(Kekre et al., 2008)		(Wang et al., 2007)	
	Capacity	PSNR	Capacity	PSNR	Capacity	PSNR	Capacity	PSNR
LEENA	48,626	41.2	35,827	59.1	35,827	59.1	786,432	37.92
BABOON	57,146	37	34,235	59.4	34,235	59.4	786,432	37.92
PEPPER	50,907	40.8	60,317	56.2	60,317	56.2	786,432	NA

Table 7. Capacity and PSNR value comparison of existing techniques (cont'd.)

Technique	PVD, LSB Replacement		Adaptive LSB, PVD		High Pay Load		Adaptive LSB Replacement	
Author	(Wu et al., 2005)		(Yang et al., 2008)		(Chen et al., 2010)		(Luo et al., 2010)	
	Capacity	PSNR	Capacity	PSNR	Capacity	PSNR	Capacity	PSNR
LEENA	774,970	37.6	807,256	41.39	774,970	37.6	807,256	41.39
BABOON	720,288	34.3	854,096	38.58	720,288	34.3	854,096	38.58
PEPPER	776,160	37.5	800,168	42.42	776,160	37.5	800,168	42.42

Table 8. Capacity and PSNR value comparison of existing techniques (cont'd.)

Technique	MPVD, Adaptive		DHPVD		Adaptive, Floor, Modulus		ADH, Modulus	
Author	(Liao et al., 2011)		(Mandal et al., 2011)		(Joo et al., 2011)		(Chen et al., 2011)	
	Capacity	PSNR	Capacity	PSNR	Capacity	PSNR	Capacity	PSNR
LEENA	810,564	39.6	NA	49.45	810,564	39.6	NA	49.45
BABOON	903,580	36.9	NA	46.54	903,580	36.9	NA	46.54
PEPPER	805,492	39.8	NA	48.78	805,492	39.8	NA	48.78

Table 9. Capacity and PSNR value comparison of existing techniques via the proposed approach

Technique	Adaptive Modulus		NPI		Color PVD		Proposed Technique	
Author	(Maleki et al., 2011)		(Imran et al., 2007)		(Mandal et al., 2012)			
	Capacity	PSNR	Capacity	PSNR	Capacity	PSNR	Capacity	PSNR
LEENA	1,055,620	34.4	1.48	42.3	145,787	42.3	806,912	60.7
BABOON	1,108,708	32.2	1.47	38.4	144,916	38.4	756,496	56.5
PEPPER	1,057,584	34.3	1.48	42.3	145,995	42.3	802,571	52.5

statistical analysis. The various results are explained below.

4.4.1. Visual Analysis

The results of the proposed technique on a 24-bit color image (FAMILY.bmp) can be seen in Figure 1. This approach successfully resisted visual attacks as visual difference in the input cover image and the stego image could not be traced as demonstrated in Figure 1.

4.4.2. Statistical Analysis: Histogram Analysis

The results of the histogram analysis are shown in Figure 2 and no differences were found in the

Figure 1. No visual differences in the original cover image and the stego image

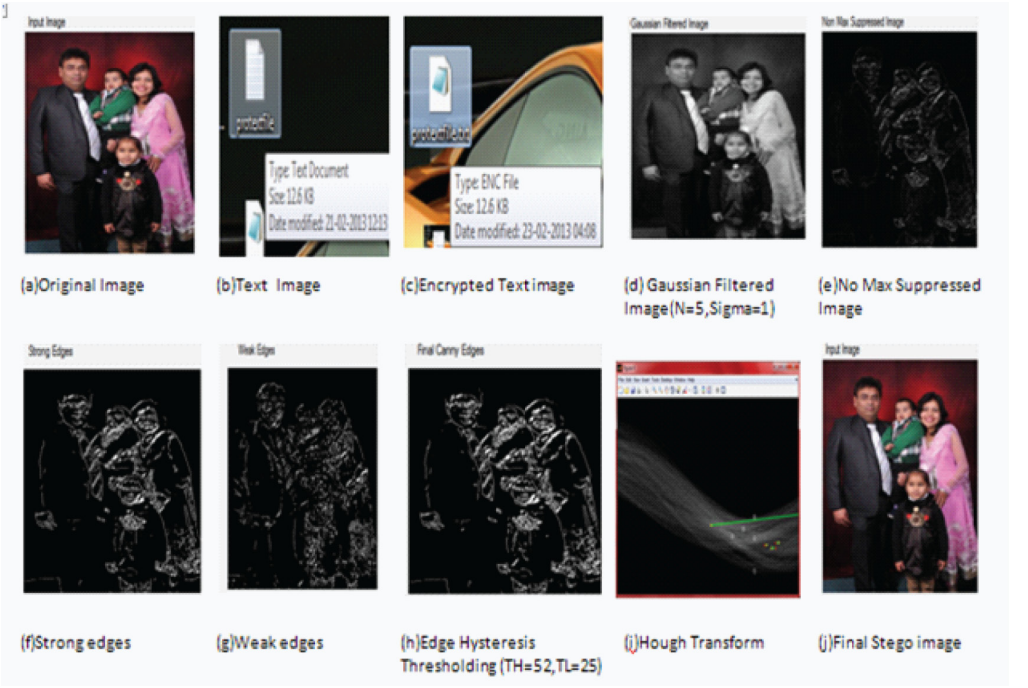
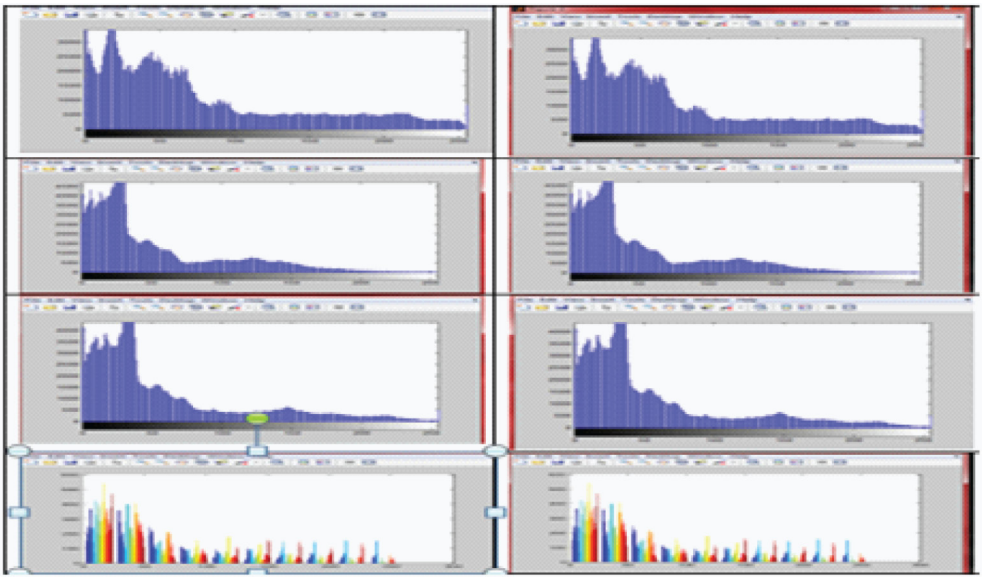


Figure 2. Histogram analysis of the red, green, blue for all components of the original image and stego image (FAMILY.bmp)



histograms of the original and stego images, so they could not be attacked.

4.4.3. Statistical Analysis:Chi-Square Attack

The proposed approach was even tested by the chi square attack and it could succesfully withstand these attacks, as shown in Figure 3.

4.4.4. Statistical Analysis RS-Analysis

The results of the RS analysis are shown in Figure 4 and the proposed approach could not be attacked by this attack. Figure 5 and Table 10 show the results of RS analysis for the family cover image. The analysis predicts that the difference between RM (Positive Regular) and R-M (Negative Regular) are less than 10%. Furthermore, the difference between SM (Positive Singular) and S-M (Negative Singular) is also less than 10%. This indicates that the image is secured.

5. CONCLUSION

This present research proposes an ideal model for covert communication using different LSB based image steganography techniques across internet. It provides high imperceptibility i.e.

Table 10. RS analysis on FAMILY.bmp

	Red	Green	Blue
Positive Regular	23.1582	23.3256	24.9753
Positive Singular	11.5232	11.3452	11.2234
Negative Regular	23.2852	24.2313	25.9876
Negative Singular	10.3221	10.2254	10.1123

Figure 3. Chi square attack on FAMILY.bmp

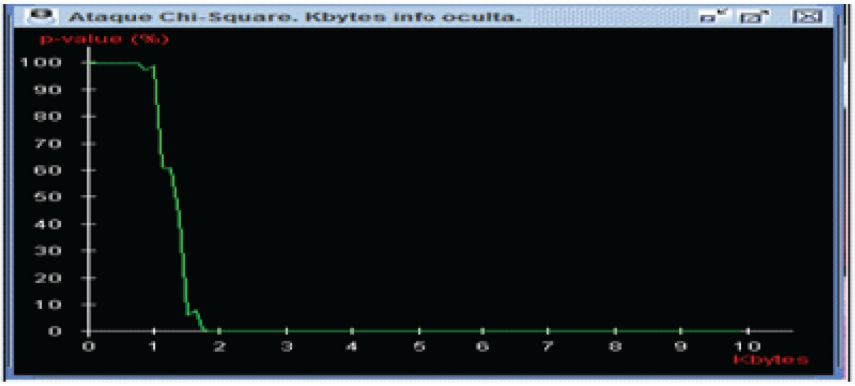


Figure 4. RS analysis of FAMILY.bmp

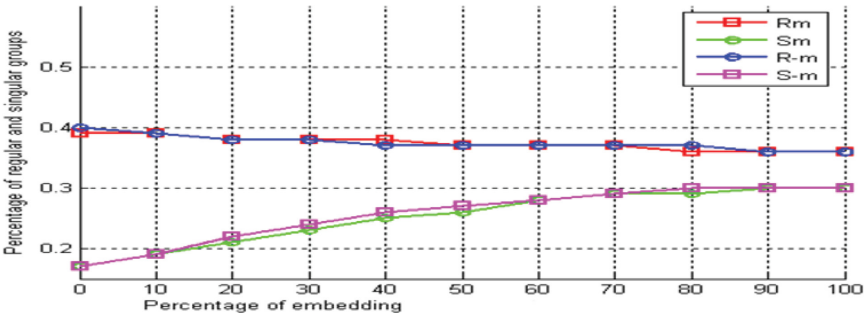
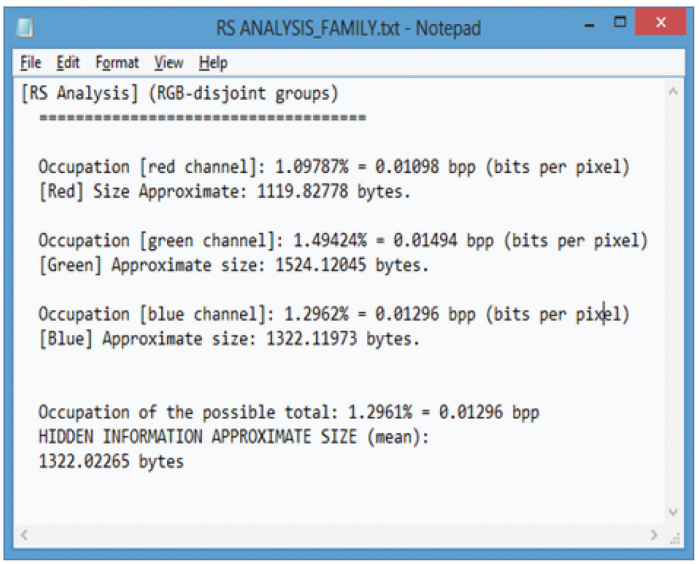


Figure 5. RS analysis of FAMILY.bmp



difficult to locate differences in cover image and stego image; provide better hiding capacity i.e. can have maximum possible data hidden in it without any degradation of image quality and robustness against various steganalysis attacks. Hybrid edge detector is proposed to divide the image in high and low transition areas efficiently. This has been achieved successfully along with additional benefits of added security and immunity to all types of disturbances like noise during communication with integration of cryptography and randomization with it. Experimental results of proposed approach are satisfactory in terms of PSNR and capacity for various test images. Comparison Analysis of output results with other existing techniques is giving the proposed approach an edge over others. The proposed approach has been thoroughly tested for various steganalysis attacks like visual analysis, histogram analysis, chi-square, and RS analysis and could sustain all these attacks very well.

REFERENCES

Alwan, R. H., Kadhim, F. J., & Al-Taani, A. T. (2005). Data embedding based on better use of bits in image pixels. *International Journal of Signal Processing*, *2*(1), 104–107.

Amirtharajan, R., Qin, J., & Rayappan, J. B. B. (2012). Random image steganography and steganalysis: Present status and future directions. *Information Technology Journal*, *11*, 566–576. doi:10.3923/itj.2012.566.576

Arjun, N. S., & Negi, A. (2006). A filtering based approach to adaptive steganography. In *Proceedings of the TENCON 2006, IEEE Region 10 Conference* (pp. 1-4).

Bassil, Y. (2012). Image steganography based on a parameterized canny edge detection algorithm. *International Journal of Computers and Applications*, *60*(4).

Canny, J. F. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *8*(6), 679–697. doi:10.1109/TPAMI.1986.4767851 PMID:21869365

Chan, C.-K., & Cheng, L. M. (2001). Improved hiding data in images by optimal moderately-significant-bit replacement. *IEE. Electronics Letters*, *37*(16), 1017–1018. doi:10.1049/el:20010714

Chan, C.-K., & Cheng, L. M. (2004). Hiding data in images by simple LSB substitution. *Pattern Recognition*, *37*, 469–474. doi:10.1016/j.patcog.2003.08.007

Chang, C.-C., Chan, C.-S., & Fan, Y.-H. (2006). Image hiding scheme with modulus function and dynamic programming strategy on partitioned pixels. *Pattern Recognition*, *39*(6), 1155–1167. doi:10.1016/j.patcog.2005.12.011

Chang, C.-C., Hsiao, J.-Y., & Chan, C.-S. (2003). Finding optimal least-significant-bit substitution in image hiding by dynamic programming strategy. *Pattern Recognition*, *36*, 1538–1595. doi:10.1016/S0031-3203(02)00289-3

Chang, C.-C., Lin, M.-H., & Hu, C. (2002). A fast and secure image hiding scheme based on LSB substitution. *International Journal of Pattern Recognition and Artificial Intelligence*, *16*(4), 399–416. doi:10.1142/S0218001402001770

Chang, C.-C., & Tseng, H. W. (2004). A steganographic method for digital images using side match. *Pattern Recognition Letters*, *25*, 1431–1437. doi:10.1016/j.patrec.2004.05.006

Chen, W., Chang, C., & Le, T. (2010). High payload steganography mechanism using hybrid edge detector. *Expert Systems with Applications*, *37*, 3292–3301. doi:10.1016/j.eswa.2009.09.050

Chou, Y.-C., Chang, C.-C., & Li, K.-M. (2008). A large payload data embedding technique for color images. *FundamentaInformaticae*, *88*(1-2), 47–61.

Ferguson, N., & Schneier, B. (2003). *Practical cryptography*. John Wiley.

Fridrich, J., Goljan, M., & Du, R. (2001). Reliable detection of LSB steganography in color and grayscale images. In *Proceedings of ACM Workshop on Multimedia and Security: New Challenges* (pp. 27-30). ACM Press.

Gutub, A., Al-Qahtani, A., & Tabakh, A. (2009, May 10-13). Triple-A: Secure RGB image steganography based on randomization. In *Proceedings of IEEE/ACS International Conference on Computer Systems and Applications (AICCSA, 2009)*, Rabat, Morocco (pp. 400-403).

Gutub, A., Ankeer, M., Abu-Ghalioun, M., Shaheen, A., & Alvi, A. (2008, March 18-20). Pixel indicator high capacity technique for RGB image based steganography. In *Proceedings of 5th IEEE International Workshop on Signal Processing and its Applications (WoSPA 2008)*, University of Sharjah, Sharjah, U.A.E.

Hempstalk, K. (2006, February 11-19). Hiding behind corners: Using edges in images for better steganography. In *Proceedings of the Computing Women's Congress*, Hamilton, New Zealand.

Hough, P. V. C. (1962). *Method and means for recognizing complex patterns*. U.S. Patent 3069654.

Hussain, M. (2010). Pixel intensity based high capacity data embedding method. In *Proceedings of the International Conference on Information and Emerging Technologies (ICIET)* (pp. 1-5).

Hussain, M., & Hussain, M. (2011, September 5-6). Embedding data in edge boundaries with high PSNR. In *Proceedings of 7th International Conference on Emerging Technologies (ICET 2011)* (pp. 1-6).

Johnson, N. F., Duric, Z., & Jajodia, S. (2000). *Information hiding, and watermarking - attacks & countermeasures*. Kluwer Academic Publishers.

Joo, J. C., Oh, T. W., Lee, H. Y., & Lee, H. K. (2011). Adaptive steganographic method using the floor function with practical message formats. *International Journal of Innovative Computing, Information, & Control*, *7*(1), 161–175.

Jung, K.-H., Ha, K.-J., & Yoo, K.-Y. (2008, August 28-30). Image data hiding method based on multi-pixel differencing and LSB substitution methods. In *Proc. International Conference on Convergence and Hybrid Information Technology (ICHIT '08)*, Daejeon, Korea (pp. 355-358).

Kahn, M. (1995). *Steganography mailing list*.

Kekre, H. B., Athawale, A., & Halarnkar, P. N. (2008). Increased capacity of information hiding in LSB's method for text and image. *International Journal of Electrical*. *Computing Systems in Engineering*, *2*(4), 246–249.

Kekre, H. B., Athawale, A., & Halarnkar, P. N. (2009). Performance evaluation of pixel value differencing and Kekre's modified algorithm for information hiding in images. In *Proceedings of the ACM International Conference on Advances in Computing, Communication and Control (ICAC3)*.

Ker, A. (2004, May 23-25). Improved detection of LSB steganography in grayscale images. In *Proc. 6th International Workshop*, Toronto, Canada(vol. 3200, pp. 97–115). Springer LNCS.

Ker, A. (2005). Steganalysis of LSB matching in gray scale images. *IEEE Signal Processing Letters*, *12*(6), 441–444. doi:10.1109/LSP.2005.847889

Kumar, P. M., & Shunmuganathan, K. L. (2012). Developing a secure image steganographic system using TPVD adaptive LSB matching revisited algorithm for maximizing the embedding rate. *Information Security Journal: A Global Perspective, 21*(2).

Lee, Y. K., & Chen, L. H. (2000). High capacity image steganographic model. *IEEE Proc., Vis. Image Signal Process, 147*(3), 288-294.

Li, B., He, J., Huang, J., & Shi, Y. Q. (2011). A survey on image steganography and steganalysis. *Journal of Information Hiding and Multimedia Signal Processing*, *2*(2).

Liao, X., Wen, Q.-Y., & Zhang, J. (2011). A steganographic method for digital images with four-pixel differencing and modified LSB substitution. *Journal of Visual Communication and Image Representation*, *22*(1), 1–8. doi:10.1016/j.jvcir.2010.08.007

Lie, W.-N., & Chang, L.-C. (1999, October 24-28). Data hiding in images with adaptive numbers of least significant bits based on the human visual system. In *Proc. IEEE Int. Conf., Image Processing*, Kobe, Japan (pp. 286-290).

Liu, J.-C., & Shih, M.-H. (2008). Generalizations of pixel value differencing steganography for data hiding in images. *Fundamenta Informaticae*, *83*(3), 319–335.

Liu, S.-H., Chen, T.-H., Yao, H.-X., & Gao, W. (2004, August 26-29). A variable depth LSB data hiding technique in images. In *Proc. 2004 International Conference on Machine Learning and Cybernetics*, Shanghai, China (vol. 7, pp. 3990-3994).

Luo, W., Huang, F., & Huang, J. (2010). Edge adaptive image steganography based on LSB matching revisited. *IEEE Trans. Inf. Forensics Security*, *5*(2), 201–214. doi:10.1109/TIFS.2010.2041812

Maleki, N., Jalali, M., & VafaeiJahan, M. (2011). An adaptive data hiding method using neighborhood pixels differencing based on modulus function. In *Proceedings of the International Conference on. Information Processing, Computer Vision, and Pattern Recognition (IPCV'11)*, Las Vegas, NV.

Mandal, J. K., & Das, D. (2012). Color image steganography based on pixel value differencing in spatial domain. [IJIST]. *International Journal of Information Sciences and Techniques*, *2*(4). doi:10.5121/ijist.2012.2408

Mandal, J. K., & Khamrui, A. (2011). A data-hiding scheme for digital image using pixel value differencing (DHPVD). In *Proceedings of the International Symposium on Electronic System Design (ISED)* (pp. 347–351).

Mielikainen, J. (2006). LSB matching revisited. *IEEE Signal Processing Letters*, *13*(5), 285–287. doi:10.1109/LSP.2006.870357

Nissar, A., & Mir, A. H. (2010). Classification of steganalysis techniques: A study. [Elsevier.]. *Digital Signal Processing*, *20*(6), 1758–1770. doi:10.1016/j.dsp.2010.02.003

Park, Y. R., Kang, H. H., Shin, S. U., & Kwon, K. R. (2005). A steganographic scheme in digital images using information of neighboring pixels. In *Proc. International Conference on Natural Computation*, Berlin, Germany(vol. 3612, pp. 962–968). Springer-Verlag LNCS.

Parvez, M. T., & Gutub, A. (2008). RGB intensity based variable-bits image steganography. In *Proceedings of 3rd IEEE Asia-Pacific Services Computing Conference (APSCC 2008)*, Yilan, Taiwan.

Provos, N., & Honeyman, P. (2002). Detecting steganographic content on the internet. In *Proceedings of NDSS'02: Network and Distributed System Security Symposium* (pp. 1-13). Internet Society.

Roque, J. J., & Minguet, J. M. (2009). SLSB: Improving the steganographic algorithm LSB. In *Proceedings of the 7th International Workshop on Security in Information Systems* (pp. 57-66).

Shariq Imran, A. YounusJaved, M., & Khattak, N. S. (2007). A robust method for encrypted data hiding technique based on neighborhood pixels information. World Academy of Science, Engineering and Technology.

Singh, M., Singh, B., & Singh, S. S. (2007). Hiding encrypted message in the features of images. *IJCSNS, 7*(4).

Singh, N., Bhati, B. S., & Raw, R. S. (2012). A novel digital image steganalysis approach for investigation. *International Journal of Computers and Applications*, *47*(12), 18–21.

Specification for the Advanced Encryption Standard (AES). (2012). *Federal information processing standards publication 197.*

Stanley, C. A. (2005). *Pairs of values and the chi-squared attack* (pp. 1–45). CiteSeer.

Svvalin, G., & Lenka, S. K. (2012). A novel approach to RGB channel based image steganography technique. *International Arab Journal of e-Technology, 2*(4).

Thien, C. C., & Lin, J. C. (2003). A simple and high-hiding capacity method for hiding digit-by-digit data in images based on modulus function. *Pattern Recognition*, *36*, 2875–2881. doi:10.1016/S0031-3203(03)00221-8

Wang, C.-M., Wu, N.-I., Tsai, C.-S., & Hwang, M.-S. (2008). A high quality steganographic method with pixel-value differencing and modulus function. *Journal of Systems and Software*, *81*(1), 150–158. doi:10.1016/j.jss.2007.01.049

Wang, R.-Z., Lin, C.-F., & Lin, J.-C. (2000). Hiding data in images by optimal moderately significant bit replacement. *IET Electronics Letters*, *36*(25), 2069–2070. doi:10.1049/el:20001429

Wang, R.-Z., Lin, C.-F., & Lin, J.-C. (2001). Image hiding by optimal LSB substitution and genetic algorithm. *Pattern Recognition*, *34*, 671–683. doi:10.1016/S0031-3203(00)00015-7

Westfeld, A., & Pitzmann, A. (1999). Attacks on steganographic systems: Breaking the steganographic utilities Ezstego, Jsteg, Steganos and s-tools-and some lessons learned. In *Proceedings of the 3rd Information Hiding Workshop* (vol. 1768, pp. 61-76). LNCS Springer.

Wu, D. C., & Tsai, W. H. (2003). A steganographic method for images by pixel-value differencing. *Pattern Recognition Letters*, *24*(9-10), 1613–1626. doi:10.1016/S0167-8655(02)00402-6

Wu, H. C., Wu, N. I., Tsai, C.-S., & Hwang, M. S. (2005). Image steganographic scheme based on pixel-value differencing and LSB replacement methods. *IEE Proceedings. Vision Image and Signal Processing*, *152*(5), 611–615. doi:10.1049/ip-vis:20059022

Xiaolong, L., Yang, B., Cheng, D., & Zeng, T. (2009). A generalization of LSB matching. *IEEE Signal Processing Letters*, *16*(2), 69–72. doi:10.1109/LSP.2008.2008947

Yang, C. H., & Weng, C. Y. (2006). A steganographic method for digital images by multi-pixel differencing. In *Proc. International Computer Symposium*, Taipei, Taiwan (pp. 831-836).

Yang, C.-H., Weng, C.-Y., Wang, S.-J., & Sun, H.-M. (2008). Adaptive data hiding in edge areas of images with spatial LSB domain systems. *IEEE Transactions on Information Forensics and Security*, *3*(3), 488–497. doi:10.1109/TIFS.2008.926097

Zhang, X., & Wang, S. (2004). Vulnerability of pixel value differencing steganography to histogram analysis and modification for enhanced security. *Pattern Recognition Letters*, *25*, 331–339. doi:10.1016/j.patrec.2003.10.014

Mamta Juneja is an Assistant Professor in University Institute of Engineering and Technology, Panjab University, Chandigarh, India. She did Doctorate from Punjab Technical University in the area of image processing and information security. She is editorial committee member of various International Journals and Conferences. She is having papers in several referred journals and international conferences. Her interest areas include Image Processing, Steganography, Information Hiding and Information Security.