# Security Building at the Line of Control for Image Stego

Rengarajan Amirtharajan
Assistant Professor / ECE
School of Electrical & Electronics Engineering
SASTRA University

Benita Bose
Department of ECE
School of Electrical & Electronics Engineering
SASTRA University
Thanjavur

Sasidhar Imadabathuni
Department of ECE
School of Electrical & Electronics Engineering
SASTRA University

John Bosco Balaguru Rayappan
Associate Dean Research
School of Electrical & Electronics Engineering
SASTRA University

## ABSTRACT
Security has virtually become an indispensable non-functional requirement for any technology that deals with data. Traditional methods of imparting security using encryption have given away themselves long ago to cryptanalytic ventures. A different method of secure data transfer without revealing the mere existence of secret, called Steganography is a promising technique to ensure security. This technique camouflages secret data into a casual cover image, without affecting its visually perceived quality. This paper proposes a novel technique, which is a hybrid of cryptography, edge detection and steganography. By differentially embedding secret data into edges and smooth pixels of cover image, so that edge pixels and smooth pixels have data with different encryptions, the cryptic effect can be boosted to a greater limit, making the unauthorized extraction of secret data impossible.

## General Terms
Information Security

## Keywords
Edge Detection, Information hiding, Least Significant Bit (LSB) Embedding, Optimal Pixel Adjustment Process (OPAP)

## 1. INTRODUCTION
Technology has evolved significantly from just an invention of smoke signals to eroding physical barriers of advanced communication and has allowed people to exchange information globally. But over a period of time as technology increased, morals and ethics slowly weakened and Information exchange became insecure. Confidential information in governments regarding state or country, business information, research information, information and dealings in financial institutions, defense information are conglomerated ,stored and are exchanged between nations for various needs and purposes. When there is a breach during the transfer of such critical information, it could lead to bankruptcy of a country, tyranny, mayhem, or attacks on a country by terrorists. Thus confidential information should be protected for social and ethical needs. The only tool that equips science to protect stored information and secure information transfer is *Information Security*. Information security plays a vital role in safeguarding information, ensuring the authenticity of the transmitted information, and maintaining the integrity of it. Experts have various definitions for information security and the most prominent one among them is "Information security is concerned with the confidentiality, integrity and availability of data regardless of the form the data may take: electronic, print, or other forms". There are several robust methods, which are the brain child of modern technology to implement information security and some of these are still under research to improve its potency in every feasible way. Information security is therefore an integral part of information exchange, without which impregnable data transfer is hard to imagine.

Furthermore in the present electronic epoch, secure communication is the most sought after asset and mankind is on a constant search for modes of communication to ensure the confidentiality of crucial information. This towering need for maintaining secrecy of critical information has given rise to several subclasses of information hiding techniques namely steganography, cryptography and watermarking techniques. Steganography is the art of camouflaging the vital information in a cover object in such a way that its very existence is concealed where the cover object may be Audio, Video or An Image [7,8, 11]. In cryptography [10], although the presence of information is conspicuous the code is indecipherable as it is sent in an inscrutable format.

In the recent past, several steganographic methods [1-9, 11-21] have been proposed in image steganography and they are classified into two major types namely spatial domain [2-6, 12-13, 19, 21] and frequency [18, 20] domain techniques. In the spatial domain the encrypted secret data is hidden in the pixels of cover image by employing Least Significant Bit (LSB) [ 2-6,12,13,19,21,], pixel value differencing [4,5,14] and Pixel Indicator [ 2-5] based schemes. These schemes have been adapted by many authors to achieve good imperceptibility with higher pay load [12,19,21]. In the frequency domain methods, the secret data is hidden in the transformed coefficients of the cover image where Discrete Cosine Transform (DCT) [1,16,18] and Discrete Wavelet Transform (DWT) [1,16,20] act as domain changers.

The technique which is extensively used in image based steganography is least significant bit (LSB) substitution technique due to its potentiality of embedding secret information in an image with high capacity and imperceptibility. In LSB embedding process, authors have adapted raster scan [1-5,7,8,10-15, 19-21] as well as random scan [6,9,16,18] to hide the secret data in the visited pixel. Between these two scans, random is preferred over raster to increase the level of complexity against eavesdroppers. But the real challenge lies in maintaining good imperceptibility of the stego-image and sharing the secret key for retrieving the original message. To do so, an edge pixel finding algorithm is employed and this is known as *Edge Detection*. The edge detection is a technique where the edges in an image are found out by using a suitable algorithm. The edge which typically has

varying gray levels, is mainly used to define the boundary regions of the image fragments and they are used to analyze the important aspects of the image. The classical edge detection algorithms include Sobel, Prewitt, Zerocrossing, Laplacian and Canny operators.

# 2. RELATED WORK

One of the straight solutions of hiding data is to directly replace LSB of each pixel in the cover image with the bits of secret data. It results in less distortion than directly manipulating the MSB's of each pixel in the cover image. It is because the MSB is more sensitive to our vision and changing the parts of MSB will gravely degrade the quality of stego image. Hence, LSB substitution method is very simple and the scheme could maintain a good quality image. The general procedures for data hiding by simple LSB substitution are described as follows.

Let C be the original 8-bit gray scale cover-image of $M_c \times N_c$ pixels represented as

$$C=\{C_{ij} \mid 0\le i<M_c, 0\le j<N_c, C_{ij} \in \{0,1\ldots.255\}\} \quad (1)$$

SM be the secret message of length n bit as

$$SM = \{sm_i | 0 \le i<n, m_i \in \{0; 1\}\} \quad (2)$$

Suppose that secret message (SM) will be embedded into the k-LSBs of the cover-image C. Initially, the secret message M is rearranged to form conceptually k-bit virtual image SM′ represented as

$$SM'=\{sm_i' | 0 \le i<n', m_i' \in \{0, 1\ldots.. 2^k - 1\}\} \quad (3)$$

Where $n' < M_c \times N_c$. The mapping between secret message SM = $\{sm_i\}$ and the embedded message SM' = $\{m_{i'}\}$ can be defined as follows:

$$sm_i' = \sum_{j=0}^{k-1} sm_{i\times k+j} \times 2^{k-1-j}$$

Furthermore, a subset of n′ pixels $\{x_{l1}, x_{l2\ldots} x_{ln'}\}$ is chosen from the cover-image C in a pseudo random sequence. The embedding process is carried out by replacing the k LSBs of $C_{i,j}$ by $sm_i'$. Mathematically, the pixel value $C_{i,j}$ of the selected pixel for storing the k-bit message $m_{i'}$ is modified to form the stego-pixel $S_{i,j}$ as follows:

$$S_{i,j} = C_{i,j} - C_{i,j} \bmod 2^k + sm_i' \quad (4)$$

In the extraction process, given the stego-image S, the embedded messages can be readily extracted by blind extraction without using the cover-image. Using the same pseudo random sequence as in the embedding process, the set of selected pixels $\{x_{l1'}, x_{l2'}\ldots x_{ln'}\}$ from the Stego cover which have the secret message bits within it. The k LSBs of those pseudo randomly selected pixels are extracted to reconstruct the secret message bits. Mathematically, the embedded message bits $m_i$ can be recovered by

$$sm'_i = S_{i,j} \bmod 2^k \quad (5)$$

if all the pixels in the cover-image are pseudo randomly used for the embedding of secret message by the simple LSB substitution method.

## 2.1 Classic Edge Detection Techniques:

### *2.1.1 Sobel's algorithm:*

An edge is reported in an image when there is a considerable change in the image's intensity. There are several techniques employed to detect an edge. Sobel is one of the primitive edge detection techniques which are successful in identifying an edge by utilizing the gradient of the image intensity. In this algorithm, the gradient of the image intensity is taken at each point on the image which gives the magnitude and direction of increase in the intensity. Thus by comparing the resultant 2D vector with the threshold values, the presence of an edge is effectively perceived. Using the "EDGE" function in matlab along with the Sobel algorithm results in a binary image with 1's corresponding to the edges in the input image and 0's elsewhere.

### *2.1.2 Prewitt Algorithm:*

Prewitt algorithm is similar to the Sobel algorithm wherein the source image point is convolved with two 3x3 kernels to obtain the horizontal and vertical derivatives. Using the "EDGE" function in matlab along with the Prewitt algorithm results in a binary image with 1's corresponding to the edges in the input image and 0's elsewhere.

### *2.1.3 Roberts Edge Detection Algorithm:*

Roberts's algorithm aims at detecting an edge in an image by computing the intensity gradient just like the Sobel and Prewitt technique. Nevertheless, this algorithm alleviates the complexity as it employs a simple 2×2 kernel for gradient computation. In addition for every pixel its gradient intensity is compared only with that of those pixels diagonally adjacent. However, its main disadvantage is that since it uses such a small kernel, it is very sensitive to noise.

### *2.1.4 Laplacian of Gaussian Algorithm (log) and Zero Crossing Edge detection Algorithm:*

This algorithm is different from the above mentioned algorithms as it does not use the first derivative and does not involve the computation of gradients unlike the other edge detection techniques, instead it makes use of the zero crossing technique. Initially, to minimize the effect of noise and to augment its smoothness the Gaussian function is employed. The laplacian of the resultant is then calculated. When there is a change in sign in the laplacian of the Gaussian, or in other words when the laplacian value crosses zero, an edge is detected and the resultant binary image has a high vavue-'1' in the corresponding point to indicate the edge.

Zero crossing also reports an edge at any place where the image intensity gradient starts increasing or starts decreasing, and this may happen at places that are not obviously edges. Hence the log method employs other alternatives for edge detection. The simplest method being utilizing threshold of the log image value so that a value higher than that would report an edge. The problem associated with this method is that there is a possibility of having multiple edges detected. Another method involves comparing the log value of a pixel with that of its adjacent pixels and choosing those points that have a log magnitude lesser than that of its four neighbouring points. However there is a risk of missing few edges in this technique.

### *2.1.5 Canny Edge Detection Algorithm:*

The canny algorithm is one of the most sophisticated edge detection techniques since it is less prone to the detrimental effects of noise, consequently resulting in a more credible binary

image. In order to eliminate the effects of noise this algorithm employs convolution of the original image with a Gaussian filter. The resultant image is then utilized to reckon the intensity gradient and the direction of the gradient is appraised based on the gradient angle. A point on the raw image is declared an edge if the intensity strength at that point is higher than that of its adjacent points along the gradient direction. Ultimately this algorithm establishes its superiority by employing two threshold values-the higher value to include the obvious genuine edges and the lower value to trace the minute details of the image. Hence the canny edge detection algorithm gives a more elaborate and detailed binary image when used in unison with the "EDGE" function in matlab.

### *2.1.6 Proposed Hybrid Edge detection method*

The hybrid method has been employed by simply using OR of all the six available methods, which intern gives more edges and hence increases the embedding capacity.

## 3. METHODOLOGY

The LSB method is preferred over the other steganographic techniques as it yields comparatively higher payload efficiency. Edge detection on the other hand is one of the most fundamental image analysis operations. Edges are often vital clues towards the analysis and interpretation of image information, both in biological vision and computer image analysis. Edges occur mainly due to the discontinuities in depth, surface orientation, scene illumination and material properties.

The results from literature have shown that putting the edge pixels effectively into usage in the LSB substitution method facilitates an increase in the payload as well as renders the stego image virtually imperceptible and impregnable.. Further in this paper we employ a combination of several edge detection algorithms namely Sobels algorithm, Prewitts algorithm, zero crossing algorithm, Roberts algorithm, log algorithm and canny algorithm. This combinational algorithm has its edge over the individual algorithms as it increases the number of edge pixels obtained in a given image. Figure 1 elucidates the Block Diagram representation of the Proposed Method.

### 3.1 Flowchart for Embedding:

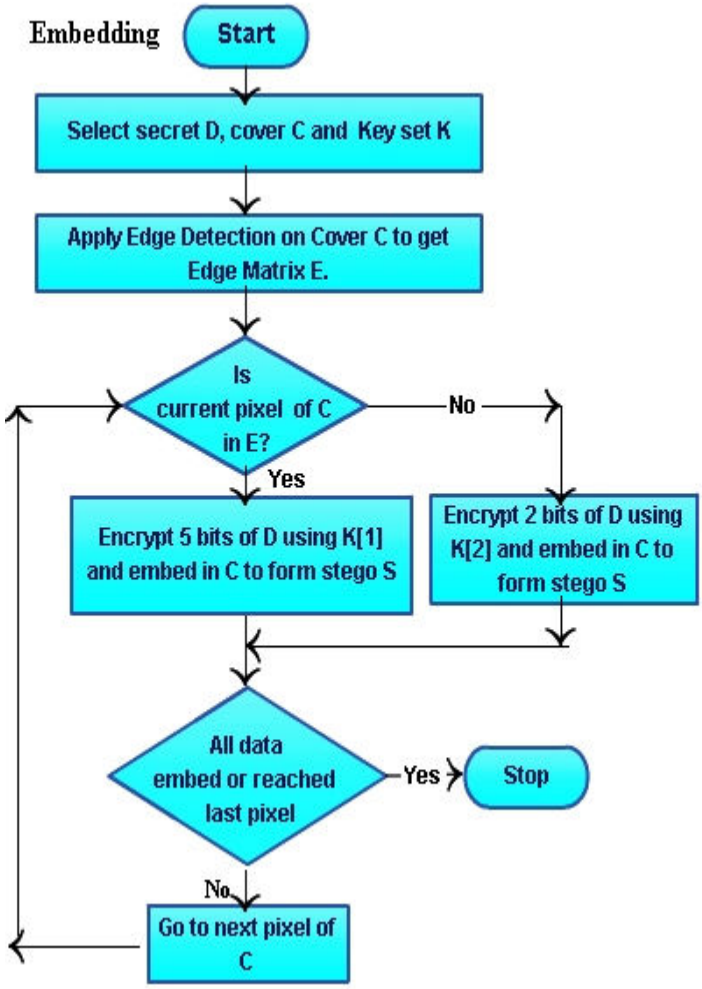



**Figure 2 Flowchart for Embedding**

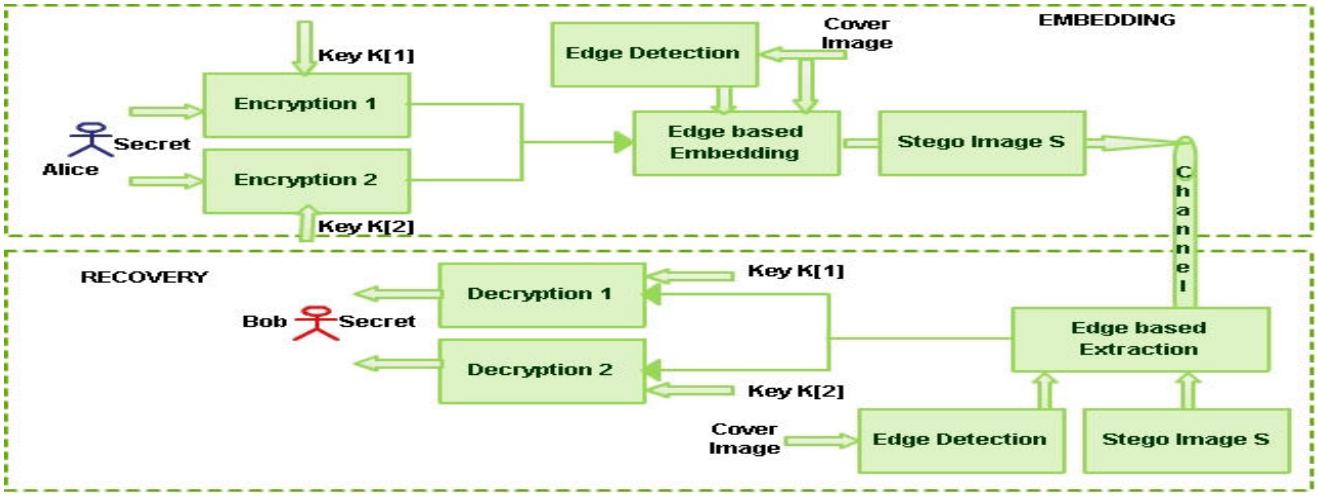



**Figure 1 Block Diagram representation of the Proposed Method**

## 3.2 Flowchart for Extraction:

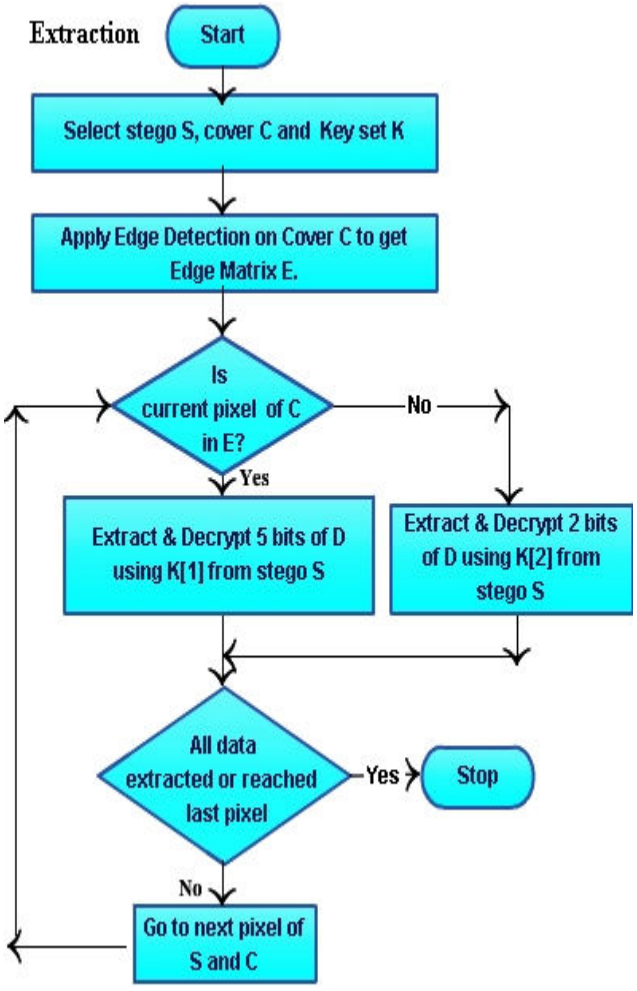



**Figure 3 Flowchart for Extraction.**

## 3.3 Embedding & Extraction Algorithm:

This methodology aims in selecting the edge pixels in the color image. In these edge pixels 5 or 6 bit LSB Substitution is performed. In order to increase the embedding capacity 1 to 2 bit LSB Substitution is performed in the smooth portions of the image (non-edge pixels). By using this methodology all the pixels in the image are utilized for embedding secret message. Since 1 to 2 bit embedding is done in the smooth region there will be minor difference between the actual image and the stego-image. The edge pixels offer high embedding capacity and hence 5 or 6 bit embedding will not distort the image quality. The resultant stego-image will be having high robustness and randomness. Due to these qualities the image will be imperceptible to the human visual system.

**Embedding Algorithm**

Step 1: Read the cover image C
Step 2: Apply Edge Detection on C to get Edge Matrix E as per the user selection.
Step 3: Read Secret Data D
Step 4: Read the Key Set K.
Step 5: For each pixel in C, do the following
Step 5.1 If (current pixel ∈ E)
Encrypt next 5 bits of D using K[1].
Embed 5 bits of D into current pixel of C.
Else
Encrypt next 2 bits of D using K[2].
Embed 2 bits of D into current pixel of C.
Step 5.2 If all data in D has been embedded
Go to Step - 6.
Step 6: Store the resulting image as Stego Image S.
Step 7: Apply OPAP on S to reduce MSE.
Step 8: Transmit S as stego image.

**Extraction Algorithm**

Step 1: Read the Stego Image S and Reference Cover C.
Step 2: Apply Edge Detection on C to get Edge Matrix E as per the user selection.
Step 3: Read the Key Set K.
Step 4: For each pixel in C, do the following
Step 4.1. If (current pixel in C ∈ E)
Recover next 5 bits of secret data from current pixel of S.
Decrypt the 5 bits of data using K[1].
Else
Recover next 2 bits of secret data from current pixel of S.
Decrypt 2 bits of data using K[2].
Step 4.2. If all data gas been recovered Go to Step - 5.
Step 5: Store the resulting data as Secret D.

## 4. RESULT & DISCUSSION

A novel steganography scheme is further discussed and thoroughly analyzed. In generality, on concealment of data in a cover image, the last bits (LSBs) of the pixels of the image are affected and embedded into. In this method, the capacity or the overall embedding payload of the system is modestly low. Thus the capacity requirement of the magic triangle is compromised upon, contrary to the expectation that a steganographic system must have high embedding payload as when compared to the robustness. To overcome this shortcoming and to improve the imperceptibility, we improve the embedding efficiency i.e. embed more data per modification to the cover data and also avoid embedding in conspicuous parts in the cover image. This is done by conflating the LSB technique with edge detection mechanism.

Edge extraction is basically done so as to discern the edge and non-edge pixels, here edges are characterised by significant dissimilarity indicating boundary, and thus are comparatively obscure to human visual perception ,facilitating more amount of data to be concealed in them. In fact, whereas only 1 or 2 bits of non edge pixels can be substituted without distortion, around 5 to 6 bits of edge pixels can be replaced without perceviability in the cover image. Furthermore, this technique which was formerly proposed majorly for gray scale images is now further exserted for color images. Thus the same logic of least significant bit insertion (LSB) is used in combination with a the proposed hybrid edge detection principle, considering the cover media to be a color image. A plethora of options for edge detectors are available (Sobel, Prewitt, Laplacian, Zero crossings, Robert, Canny etc.) and they are prominently preferred for their exemplary characteristics especially since clear and sharp edges are obtained with less computational efforts. The proposed methodology uses a combinational cutting edge technology of all the fore mentioned edge detectors known as hybrid edge detection algorithm and it incorporates the advantages of all the six edge

detection techniques. As a result, numerous edge pixels are obtained and thus more data can be embedded, increasing the payload. A given color image can be furcated into three components, RED GREEN and BLUE, every component is then in behavioral characteristics similar to a grey scale image. Then every pixel of the individual components is further worked upon. The component is then filtered using the hybrid edge detector(combination of all six classic edge detection methods )upon which we obtain many more edge pixels of the component than if either were used individually. The pixels are then further ramified into edge and non-edge pixels, if there are n pixels- P1, P2.....Pn, then the status of the pixels P2 to Pn are calculated and stacked away in the P1 value's last bits. The maximum value should be (n-1) bits that should be accommodated in P1. To preserve the quality of pixel P1 as well as to increase the embedding payload the general values of n are taken as 3, 4 or 5. On classification, each non-edge pixel of cover image can be embedded using LSB and the data to be embedded maybe restricted to 1 or 2 to preserve the imperceptibility, however in the edge pixels, around 3,4 or 5 bits can be replaced without any perceptible distortion. And thus all the pixels of the cover image are utilized, and since only 1 to 2 bits of smooth pixels are embedded, the difference between the actual and stego-image is inconspicuous, with the edge pixels offering high payload, since embedding in 5 to 6 bits also doesn't have a major aberration. Thus the stego image obtained has high imperceptibility to the human visual system attributed to high payload commingled with eminent robustness and randomness. The image show considerable resistance to steganalysis and the randomness can be further fortified by combining LSB substitution with Raster scan technique.

In this present implementation Lena, baboon, Gandhi and Temple of 256 × 256 color digital images has been taken as cover images as shown in Figure 4 a, b, c & d and tested for full embedding capacity and the results are given. The effectiveness of the stego process proposed has been studied by calculating MSE and PSNR for all the four digital images in RGB planes using the proposed method.

The MSE is calculated by using the equation,

$$MSE=\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(X_{i,j}-Y_{i,j}\right)^2 \qquad (6)$$

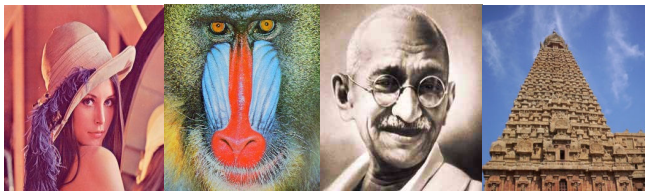
where $X_{i,j}$ is Stego value and $Y_{i,j}$ is the cover object.

The PSNR is calculated using the equation

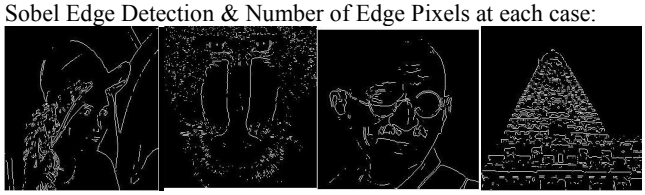
$$PSNR = 10\log_{10}\left(\frac{I_{\max}^2}{MSE}\right)dB \qquad (7)$$

where $I_{max}$ is the intensity value of each pixel which is equal to 255 for 8 bit gray scale images. Higher the value of PSNR better the image quality.The corresponding results of MSE, PSNR. No of Edge Pixels, Total number of Edge pixels in RGB and the total payload for each case is given in Table 1 to Table 8. **Results prove that the proposed edge detection based steganography supersedes all the methods**

Figure 4 a 4b 4c 4d

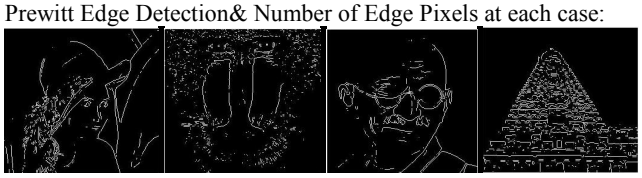


Sobel Edge Detection & Number of Edge Pixels at each case:



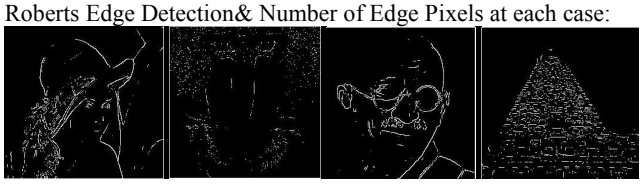
7977 8277 6084 13329

Prewitt Edge Detection& Number of Edge Pixels at each case:



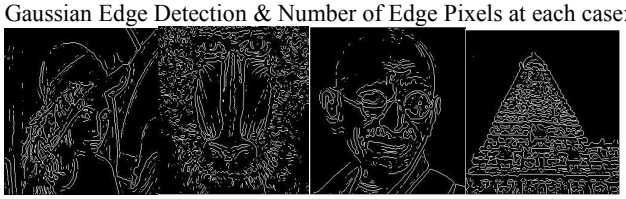
7848 7933 6093 13000

Roberts Edge Detection& Number of Edge Pixels at each case:



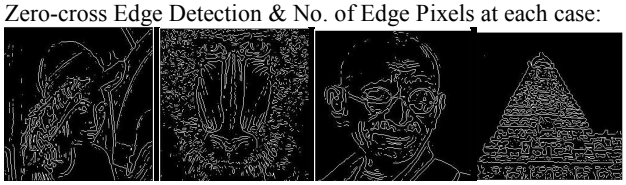
6916 3431 6170 8344

Gaussian Edge Detection & Number of Edge Pixels at each case:



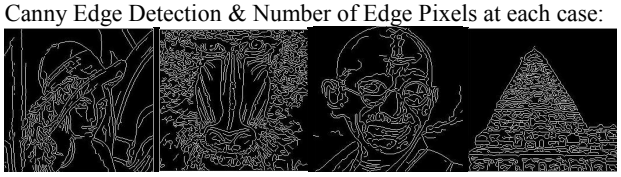
13197 22329 11214 19449

Zero-cross Edge Detection & No. of Edge Pixels at each case:



13197 22329 11214 19449

Canny Edge Detection & Number of Edge Pixels at each case:



16841 31697 14736 22250

The Proposed Edge Detection & No. of Edge Pixels at each case:

**29704 49816 25520 40769**

**Table 1 : Lena Edge Pixels k1=5, and Remaining Smooth Pixels k2=1**

| Edge Detection Method | | Sobel | Prewitt | Roberts | Log | Zerocross | Canny | Hybrid |
|---|---|---|---|---|---|---|---|---|
| MSE | Red | 2.4245 | 2.4004 | 2.0689 | 3.9045 | 3.9045 | 4.6863 | **8.3952** |
| | Green | 2.2881 | 2.295 | 2.0275 | 3.518 | 3.518 | 4.6526 | **7.8346** |
| | Blue | 2.3863 | 2.3679 | 2.1256 | 4.2273 | 4.2273 | 5.4306 | **8.8173** |
| PSN R | Red | 44.2845 | 44.3279 | 44.9734 | 42.2151 | 42.2151 | 41.4225 | **38.8905** |
| | Green | 44.536 | 44.523 | 45.0612 | 42.6679 | 42.6679 | 41.4539 | **39.1906** |
| | Blue | 44.3535 | 44.3871 | 44.856 | 41.8702 | 41.8702 | 40.7823 | **38.6775** |
| No of Edge Pixels | Red | 2671 | 2625 | 2337 | 4383 | 4383 | 5388 | **9803** |
| | Green | 2554 | 2502 | 2318 | 4070 | 4070 | 5230 | **9225** |
| | Blue | 2752 | 2721 | 2261 | 4744 | 4744 | 6223 | **10676** |
| | Total in RGB | 7977 | 7848 | 6916 | 13197 | 13197 | 16841 | **29704** |
| Bits embedded | | **228516** | **228000** | **224272** | **249396** | **249396** | **263972** | **315424** |

**Table 2 Lena Edge Pixels k1=6, and Remaining Smooth Pixels k2=2**

| Edge Detection Method | | Sobel | Prewitt | Roberts | Log | Zerocross | Canny | Hybrid |
|---|---|---|---|---|---|---|---|---|
| MSE | Red | 9.8552 | 9.2795 | 8.514 | 14.6597 | 14.6597 | 17.9991 | **32.7513** |
| | Green | 9.0347 | 8.9982 | 8.4107 | 13.5984 | 13.5984 | 16.6744 | **29.6416** |
| | Blue | 9.9491 | 9.4514 | 7.9554 | 15.8787 | 15.8787 | 20.0474 | **35.1875** |
| PSNR | Red | 38.1941 | 38.4555 | 38.8295 | 36.4696 | 36.4696 | 35.5783 | **32.9785** |
| | Green | 38.5717 | 38.5892 | 38.8825 | 36.7959 | 36.7959 | 35.9103 | **33.4118** |
| | Blue | 38.153 | 38.3758 | 39.1242 | 36.1226 | 36.1226 | 35.1102 | **32.6669** |
| No of Edge Pixels | Red | 2671 | 2625 | 2337 | 4383 | 4383 | 5388 | **9803** |
| | Green | 2554 | 2502 | 2318 | 4070 | 4070 | 5230 | **9225** |
| | Blue | 2752 | 2721 | 2261 | 4744 | 4744 | 6223 | **10676** |
| | Total in RGB | 7977 | 7848 | 6916 | 13197 | 13197 | 16841 | **29704** |
| Bits embedded | | **425124** | **424608** | **420880** | **446004** | **446004** | **460580** | **512032** |

**Table 3 Baboon Edge Pixels k1=5, and Remaining Smooth Pixels k2=1**

| Edge Detection Method | | Sobel | Prewitt | Roberts | Log | Zerocross | Canny | Hybrid |
|---|---|---|---|---|---|---|---|---|
| MSE | Red | 2.4794 | 2.5089 | 1.1253 | 6.3828 | 6.3828 | 8.8192 | **14.0238** |
| | Green | 2.6405 | 2.4577 | 1.1227 | 6.6318 | 6.6318 | 9.1104 | **14.4759** |
| | Blue | 2.3004 | 2.2128 | 1.1751 | 6.0303 | 6.0303 | 8.8084 | **14.0771** |
| PSNR | Red | 44.1874 | 44.1359 | 47.6181 | 40.0807 | 40.0807 | 38.6765 | **36.6621** |
| | Green | 43.9139 | 44.2255 | 47.6281 | 39.9145 | 39.9145 | 38.5354 | **36.5244** |
| | Blue | 44.5128 | 44.6814 | 47.43 | 40.3274 | 40.3274 | 38.6818 | **36.8348** |
| No of Edge Pixels | Red | 2807 | 2697 | 1170 | 7598 | 7598 | 10318 | **16726** |
| | Green | 2907 | 2784 | 1149 | 7696 | 7696 | 10782 | **17048** |
| | Blue | 2563 | 2452 | 1112 | 7035 | 7035 | 10597 | **16042** |
| | Total in RGB | 8277 | 7933 | 3431 | 22329 | 22329 | 31697 | **49816** |
| Bits embedded | | **229716** | **228340** | **210332** | **285924** | **285924** | **323396** | **395872** |

**Table 4 Baboon Edge Pixels k1=6, and Remaining Smooth Pixels k2=2**

| Edge Detection Method | | Sobel | Prewitt | Roberts | Log | Zerocross | Canny | Hybrid |
|---|---|---|---|---|---|---|---|---|
| MSE | Red | 10.1389 | 9.561 | 4.8367 | 25.027 | 25.027 | 34.5293 | **55.2313** |
| | Green | 10.0723 | 9.9132 | 4.7437 | 26.0759 | 26.0759 | 35.5039 | **54.3232** |
| | Blue | 10.1023 | 9.6288 | 4.8821 | 25.828 | 25.9182 | 35.0912 | **55.0198** |
| PSN R | Red | 38.0709 | 38.3258 | 41.2854 | 34.1467 | 34.1467 | 32.7489 | **30.709** |
| | Green | 38.0995 | 38.1687 | 41.3696 | 33.9684 | 33.9684 | 32.628 | **31.683** |
| | Blue | 38.0818 | 38.2234 | 41.2987 | 33.512 | 33.9123 | 32.9876 | **30.9976** |
| No of Edge Pixels | Red | 2807 | 2697 | 1170 | 7598 | 7598 | 10318 | **16726** |
| | Green | 2907 | 2784 | 1149 | 7696 | 7696 | 10782 | **17048** |
| | Blue | 2563 | 2452 | 1112 | 7035 | 7035 | 10597 | **16042** |
| | Total in RGB | 8277 | 7933 | 3431 | 22329 | 22329 | 31697 | **49816** |
| Bits embedded | | **426324** | **424948** | **406940** | **482532** | **482532** | **520004** | **592480** |

**Table 5 Mahatmagandhi Edge Pixels k1=5, and Remaining Smooth Pixels k2=1**

| Edge Detection Method | | Sobel | Prewitt | Roberts | Log | Zerocross | Canny | Hybrid |
|---|---|---|---|---|---|---|---|---|
| MSE | Red | 1.8173 | 1.8583 | 1.8744 | 3.2047 | 3.2047 | 4.0294 | **7.14** |
| | Green | 1.8552 | 1.8877 | 1.9231 | 3.2542 | 3.2542 | 4.3722 | **7.5015** |
| | Blue | 1.9516 | 1.8524 | 1.8302 | 3.3302 | 3.3302 | 4.5145 | **7.4124** |
| PSN R | Red | 45.5365 | 45.4396 | 45.4022 | 43.073 | 43.073 | 42.0784 | **39.5938** |
| | Green | 45.4468 | 45.3715 | 45.2908 | 43.0064 | 43.0064 | 41.7238 | **39.3793** |
| | Blue | 45.2269 | 45.4535 | 45.5059 | 42.9061 | 42.9061 | 41.5847 | **39.4312** |
| No of Edge Pixels | Red | 2025 | 2031 | 2070 | 3701 | 3701 | 4608 | **8233** |
| | Green | 2020 | 2031 | 2057 | 3721 | 3721 | 5007 | **8594** |
| | Blue | 2039 | 2031 | 2043 | 3792 | 3792 | 5121 | **8693** |
| | Total in RGB | 6084 | 6093 | 6170 | 11214 | 11214 | 14736 | **25520** |
| **Bits embedded** | | **220944** | **220980** | **221288** | **242264** | **242264** | **255552** | **298688** |

**Table 6 Mahatmagandhi Edge Pixels k1=6, and Remaining Smooth Pixels k2=2**

| Edge Detection Method | | Sobel | Prewitt | Roberts | Log | Zerocross | Canny | Hybrid |
|---|---|---|---|---|---|---|---|---|
| MSE | Red | 7.4439 | 7.4016 | 7.7012 | 12.6105 | 12.6105 | 14.8093 | **26.4473** |
| | Green | 7.5753 | 7.5701 | 7.6406 | 12.5555 | 12.5555 | 16.8832 | **28.0563** |
| | Blue | 7.4976 | 7.4789 | 7.6945 | 12.2753 | 12.2678 | 15.986 | **27.8276** |
| PSNR | Red | 39.4128 | 39.4375 | 39.2652 | 37.1235 | 37.1235 | 36.4254 | **33.907** |
| | Green | 39.3368 | 39.3398 | 39.2995 | 37.1425 | 37.1425 | 35.8562 | **33.6505** |
| | Blue | 39.2873 | 39.1098 | 39.2874 | 37.1987 | 37.1387 | 35.986 | **33.7654** |
| No of Edge Pixels | Red | 2025 | 2031 | 2070 | 3701 | 3701 | 4608 | **8233** |
| | Green | 2020 | 2031 | 2057 | 3721 | 3721 | 5007 | **8594** |
| | Blue | 2039 | 2031 | 2043 | 3792 | 3792 | 5121 | **8693** |
| | Total in RGB | 6084 | 6093 | 6170 | 11214 | 11214 | 14736 | **25520** |
| **Bits embedded** | | **417552** | **417588** | **417896** | **438872** | **438872** | **452160** | **495296** |

**Table 7 Temple Edge Pixels k1=5, and Remaining Smooth Pixels k2=1**

| Edge Detection Method | | Sobel | Prewitt | Roberts | Log | Zerocross | Canny | Hybrid |
|---|---|---|---|---|---|---|---|---|
| MSE | Red | 3.9047 | 3.7881 | 2.4824 | 5.5578 | 5.5578 | 6.4141 | **11.5035** |
| | Green | 3.8742 | 3.8867 | 2.5566 | 5.699 | 5.699 | 6.3831 | **11.4871** |
| | Blue | 3.874 | 3.7238 | 2.3832 | 5.6497 | 5.6497 | 6.3546 | **11.6128** |
| PSN R | Red | 42.2149 | 42.3466 | 44.182 | 40.6818 | 40.6818 | 40.0595 | **37.5225** |
| | Green | 42.249 | 42.2349 | 44.0541 | 40.5728 | 40.5728 | 40.0805 | **37.5287** |
| | Blue | 42.2492 | 42.4209 | 44.3592 | 40.6106 | 40.6106 | 40.0999 | **37.4814** |
| No of Edge Pixels | Red | 4406 | 4277 | 2792 | 6417 | 6417 | 7479 | **13431** |
| | Green | 4476 | 4356 | 2831 | 6519 | 6519 | 7452 | **13689** |
| | Blue | 4447 | 4367 | 2721 | 6513 | 6513 | 7319 | **13649** |
| | Total in RGB | 13329 | 13000 | 8344 | 19449 | 19449 | 22250 | **40769** |
| **Bits embedded** | | **249924** | **248608** | **229984** | **274404** | **274404** | **285608** | **359684** |

**Table 8 Temple Edge Pixels k1=6, and Remaining Smooth Pixels k2=2**

| Edge Detection Method | | Sobel | Prewitt | Roberts | Log | Zerocross | Canny | Hybrid |
|---|---|---|---|---|---|---|---|---|
| MSE | Red | 14.7265 | 14.658 | 10.2526 | 22.0611 | 22.0611 | 25.601 | **45.3655** |
| | Green | 15.3024 | 14.6307 | 9.539 | 20.254 | 20.254 | 23.8691 | **43.3758** |
| | Blue | 14.5465 | 14.6723 | 10.201 | 21.234 | 21.768 | 24.752 | **44.342** |
| PSNR | Red | 36.4498 | 36.47 | 38.0225 | 34.6945 | 34.6945 | 34.0482 | **31.5635** |
| | Green | 36.2832 | 36.4781 | 38.3358 | 35.0657 | 35.0657 | 34.3524 | **31.7583** |
| | Blue | 36.786 | 36.245 | 38.421 | 35.0112 | 34.987 | 34.236 | **31.635** |
| No of Edge Pixels | Red | 4406 | 4277 | 2792 | 6417 | 6417 | 7479 | **13431** |
| | Green | 4476 | 4356 | 2831 | 6519 | 6519 | 7452 | **13689** |
| | Blue | 4447 | 4367 | 2721 | 6513 | 6513 | 7319 | **13649** |
| | Total in RGB | 13329 | 13000 | 8344 | 19449 | 19449 | 22250 | **40769** |
| **Bits embedded** | | **446532** | **445216** | **426592** | **471012** | **471012** | **482216** | **556292** |

## 5. CONCLUSION

The complexity of data retrieval is greatly improved by differential embedding. The choice of two different encryption methods for edges and smooth pixels makes it impossible to compromise this system. These advantages are backed up by choice of different number of bits embedded in each pixel type. Thus, this trio of differential embedding, alternating encryption and varying depth of embedding makes a self-sufficient, reliable and robust security system. Results prove that the proposed edge detection based steganography supersedes all the methods.

## 6. ACKNOWLEDGMENTS

Our thanks to G.Vivek, Venkata Abhiram Murarisetty, Motamarri Abhilash Swarup, Mohammed shakeel shaik, Sandeep Kumar Behera and G.Aishwarya ECE Stego group Students /SEEE/ SASTRA University for their technical support.

## 7. REFERENCES

[1] Abbas Cheddad, Joan Condell, Kevin Curran, Paul Mc Kevitt, Digital image steganography: Survey and analysis of current methods Signal Processing 90 (2010) 727–752.

[2] Adnan Gutub, "Pixel Indicator Technique for RGB Image Steganography", *Journal of Emerging Technologies in Web Intelligence (JETWI)* (2010 )2(1) 56-64

[3] R.Amirtharajan, Aishwarya G, Madhumita Rameshbabu and John Bosco Balaguru Rayappan, "Optimum Pixel & Bit location for Colour Image Stego- A Distortion Resistant Approach. *International Journal of Computer Applications(2010)* 10(7):17–24.

[4] R.Amirtharajan, Adharsh.D, Vignesh.V and R.John Bosco Balaguru, "PVD Blend with Pixel Indicator - OPAP Composite for High Fidelity Steganography" *International Journal of Computer Applications (2010)* 7(9):31–37.

[5] R.Amirtharajan, Sandeep Kumar Behera, Motamarri Abhilash Swarup, Mohamed Ashfaaq K and John Bosco Balaguru Rayappan, " Colour Guided Colour Image Steganography" *Universal Journal of Computer Science and Engineering Technology 1 (1) (2010), 16-23*.

[6] R.Amirtharajan and Dr. R. John Bosco Balaguru, "*Tri-Layer Stego for Enhanced Security – A Keyless Random Approach*" - IEEE Xplore, DOI, 10.1109/IMSAA.2009.5439438.

[7] R.Amirtharajan, R. Akila, P.Deepikachowdavarapu, "A Comparative Analysis of Image Steganography". International Journal of Computer Applications 2(3)(2010):41–47.

[8] R.Amirtharajan, Krishnendra Nathella and J Harish, "Info Hide – A Cluster Cover Approach" International Journal of Computer Applications 3(5)(2010) 11–18.

[9] R.Amirtharajan and R.John Bosco Balaguru. "Constructive Role of SFC & RGB Fusion versus Destructive Intrusion". International Journal of Computer Applications 1(20):30–36

[10] Bruice Schneier, Applied Cryptography Protocols, Algorithm and Source Code in C. Second edition. Wiley India edition 2007

[11] W. Bender, D. Gruhl, N. Morimoto, A. Lu, "Techniques for data hiding" IBM Syst. J. 35 (3&4) (1996) 313–336.

[12] C.K. Chan, L.M. Chen, Hiding data in images by simple LSB substitution, Pattern Recognition 37 (3) (2004) 469–474.

[13] 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 (July), 1583–1595.

[14] Chang, C.C., Tseng, H.W., 2004. A steganographic method for digital images using side match. Pattern Recognition Letter 25 (September), 1431–1437.

[15] F.A.P. Petitcolas, R.J. Anderson, M.G. Kuhn, Information hiding—a survey, Proc. IEEE 87 (7) (1999) 1062–1078.

[16] S. Katzenbeisser, F.A.P. Petitcolas, Information Hiding Techniques for Steganography and Digital Watermarking, Artech House, Norwood, MA, 2000.

[17] Peter Wayner, "Disappearing cryptography: information hiding : steganography & watermarking" 2nd. ed. San Francisco: Morgan Kaufmann; 2002.

[18] N. Provos and P. Honeyman, "Hide and seek: An introduction to steganography," *IEEE Security Privacy Mag.*,1 (3) (2003) 32–44

[19] C.C. Thien, J.C. Lin, A simple and high-hiding capacity method for hiding digit-by-digit data in images based on modulus function, Pattern Recognition 36 (11) (2003) 2875–2881

[20] Po-Yueh Chen Hung-Ju Lin, "A DWT Based Approach for Image Steganography**",** International Journal of Applied Science and Engineering 4(3)(2006): 275-290

[21] R.Z. Wang, C.F. Lin, J.C. Lin, Image hiding by optimal LSB substitution and genetic algorithm, Pattern Recognition 34 (3) (2000) 671–683