Enhanced Chaos based Image Steganography using Edge Adaptive and Cat Mapping Techniques

V. Lokeswara Reddy
Associate professor
Department of CSE
K.S.R.M. College of Engineering
Kadapa, Y.S.R. District., A.P, India

B. Sailendar
M.Tech Student
Department of CSE
K.S.R.M. College of Engineering
Kadapa, Y.S.R. District., A.P, India

ABSTRACT
Steganography is the science of hiding the messages in a medium. The existence of hidden messages remains imperceptible to the Intruders. The medium used for hiding the messages may be audio, video, text or images. The important goal of steganography is to protect the hidden message. There are many techniques to hide the payload in images. This will be achieved by applying different techniques in any medium. There are many techniques applied by the intruders to find the hidden information those techniques are called anti-steganalysis techniques. To provide the security to the payload after inserting the payload in to messages we need to do the anti steganalysis, if the hidden information is found then we need to provide the additional security. There is a need to build the system with highest security levels so that the anti steganalysis techniques can't find out the hidden information. In the proposed system there are certain areas which are suitable for hiding the payload are called Edges. Edges are good Regions of Interest or ROIs that are used for steganography. The proposed system uses edge adaptive image steganography [1] that uses the combinations of chaotic cat mapping [2] to provide additional security and matrix encoding [3] and LSBM [4] to embed the data in to image. The proposed mechanism guaranties the high imperceptibility and Fidelity which are the two important requirements for any Image steganography.

Keywords
Payload, Chaotic cat mapping, Matrix Encoding, LSB Matching, Regions of Interests (ROI), Edge Detection, Image Steganography;

1. INTRODUCTION
The use of Internet is gradually increasing day by day. Most of the people using the internet for communicating and transferring huge amount of data through the internet. While transferring the data confidentiality of information is very important. Data may be stolen, interchanged or even destroyed by the intruders, due to this data loss, data leakages and data damages Occurs [5]. So providing confidentiality and privacy during data transfer is very important over the internet. To provide confidentiality and privacy of important data over the internet it must provides envelop such that its contents are visible to the intended receivers. Data hiding techniques are used to perform the task such as Steganography.

Steganography is the science of hiding the information into objects such that the existence of information remains imperceptible to external adversary. The objects are called cover medium and hidden information is called payload. The cover medium may be text files, audio files, video files and images. The mostly used cover medium is images because of its pervasiveness in daily applications and highly redundant in nature.

Image steganography are classified as transform and spatial domains. Transform domain is applied after transformation of pixel values and then process is applied. Spatial domain where as in spatial domain processing is applied directly on the pixel values. In the proposed mechanism spatial domain is used because of its simplicity and advantages. The advantages of spatial domain are simplicity, reducing hardware requirements, implementation time is less and low time complexity [6]. Image steganography mainly focuses on three things where to hide the information, how safe is the embedding and how secure is the payload in case of image steganography to an adversary. To all the things the proposed mechanism includes the anti-steganalysis test to provide the security so that the hidden message is not found in case of applying any anti steganalysis test. There are many existing mechanisms in Image Steganography which exhibits different implementations. Embedding the payload into the regions of interest (ROI) in a image is a new approach to image steganography. ROIs are identified as Edges because when payload is inserted into the edges in the images that produce the less amount of distortion. Human visual system is not able to notice or identify the changes made to the image after inserting the payload in to the image in edges. Embedding the payload into the random pixel position scatters the payload throughout the cover image and reduces the probability of detection by intruders.

There are various approaches for providing the security to the payload. Common approach includes pre-encrypting the payload or distorting in a pseudo random manner using various mathematical transformations and mapping is used. In the proposed mechanism ROI based spatial domain steganography mechanism is used for embedding the data in to edges of a cover image. To provide additional layer of security to the payload it is subjected to a pre embedding distortion is done by using chaotic cat mapping technique. The additional layer of security ensures the actual payload is not get revealed to the intruders even though they applied any anti-steganalysis mechanisms. Chaos based edge adaptive image steganography is used to find the payload [7] .

2. PROPOSED MECHANISM
The proposed mechanism works on spatial domain and uses Canny's Edge Detection algorithm [1] for locating the Regions of Interest in images that are Edges. The Canny Edge Detection Algorithm is the best algorithm in finding the true weak edges. This algorithm is considered as optimized and standard method for detecting edges in images as compared to other edge detection techniques [8, 9]. Pixel randomization is

necessary because the edges are scattered in the image. Then the payload scrambling algorithm is used to scramble the payload across the Edges. Embedded algorithm is used to embed the payload into the cover image. This embedded algorithm uses the combination Matrix Encoding and LSB mapping to embed the distorted payload into the cover image to generate the stego-image. The extraction of payload is done exactly reverse to the embedded algorithm. First the stego-image is taken as input to find the Edges. After finding the edges by using Matrix Encoding and LSB Matching payload is extracted.

2.1 Canny Edge Detection algorithm

The Canny's Edge Detection Algorithm is having the following steps to find the Edges in image.

The Canny Edge Detection algorithm runs in 5 steps:

Step 1: Smoothing.
- Blurring of the image to remove noise.

Step 2: Finding Gradients.
- The edges should be marked where the gradients of the image has large magnitudes.

Step 3: Non-Maximum Suppression.
- Only local maxima should be marked as edges.

Step 4: Double Thresholding.
- Potential edges are determined by thresholding.

Step 5: Edge Tracing by Hysteresis.
- Final edges are determined by suppressing all edges that are not connected to a very certain (strong) edge.

Input: Cover Image.
Output: Edges founded in Image.

To increase the security to the payload the additional layer of security is provided by applying the chaotic cat mapping [10] is applied to distort the pay load initially. The advantage of chaotic cat mapping is when it is applied after some certain number of iterations the mapping returns the original payload. This cat mapping is applied to the payload by taking number of rotations as input and generates the resultant payload. After generating the payload we are inserting the generated payload in to the Edges of the image.

$$M\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \bmod 1 \qquad (1)$$

Here $M\begin{bmatrix} x \\ y \end{bmatrix}$ gives the Chaotic cat mapping transformation over the original pixels x and y.

$$M\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & p1 \\ p2 & p1p2+1 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} \bmod N \qquad (2)$$

Where p1, p2€Z and p1, p2>=1.

The values of p1, p2 may be altered to increase the number of iterations required to bring the original payload. The proposed mechanism applies cat mapping in two steps the first step is performed before embedding. The payload is iterated k times and n is the number of iterations required to get the original payload. The second step is performed at the time of extracting the payload from the stego-image.

The embedding is performed using a combination of LSB Matching and Matrix Encoding [20]. Matrix Encoding ensures that the payload is embedded into the cover image with minimum number of pixel changes. LSB matching alleviates the pair of Value (POV) effect of the LSB Replacement Method.

Image consists of RGB Pixels. RGB values in each and every Pixel ranges from 0 to 255. The RGB pixels of image will be converted into binary values. The Edges are identified in the image by using canny's edge detection algorithm. The edges will also contain the RGB pixels, those pixels are converted into binary values. The proposed mechanism searches for the Least Significant Bit which will match for the distorted payload.

The Proposed Mechanism Embeds the payload by combining the two approaches. The combined approach work as follows

Let p1, p2, p3 be the three modifying bit position belongs to Red (R), Green (G), Blue (B) respectively and x1 and x2 are the two message bits which needs to be embedded.
Now consider

$$x1 = p1 \oplus p3 \qquad (3)$$

$$x2 = p2 \oplus p3 \qquad (4)$$

All necessary actions that need to be taken during embedding

Table 1: Action List for Embedded Conditions.

Possibilities	Appropriate Action to be taken
$x1 = p1 \oplus p3$ $x2 = p2 \oplus p3$	No change Required
$x1 \neq p1 \oplus p3$ $x2 = p2 \oplus p3$	Change component R to match condition(3) & (4)
$X1 = p1 \oplus p3$ $x2 \neq p2 \oplus p3$	Change component G to match condition(3) & (4)
$x1 \neq p1 \oplus p3$ $x2 \neq p2 \oplus p3$	Change component B to match condition(3) & (4)

The proposed mechanism consists of two phases namely Phase I and Phase II. After finding the Edges by using Canny's algorithm then first Phase performs embedding by using matrix encoding and LSB matching and second Phase performs de embedding to extract the hidden information from the cover image.

2.2 PHASE I

2.2.1 Payload scrambling algorithm

This algorithm is used scramble the payload across the edges. This algorithm works by finding the k^{th} iteration of cat mapping on the input image.

Algorithm:

Step 1: Find the number of Cat map transforms required to distort the payload and regenerate the original again. Let it be denoted by p. At each iteration i (i<p), find the two dimensional correlation coefficient between S and the output of the corresponding iteration. Store correlation coefficients in an array A.

Step 2: Find index of minimum (A). Let it be denoted by kmax.

Step 3: Set rem:=p-kmax. Perform cat map transform kmax times on S to generate the distorted payload T.

Input: Original payload S
Output: Scrambled payload T

This algorithm finds the number of cat map transformations required to distort the payload and regenerate the payload. Let p be the original payload. The payload is converted into binary bits. After converting the payload into binary bits cat mapping transformation is applied on the binary bits and the scrambled payload is denoted by T. The payload scrambling algorithm is used do scatter the payload in to the edges. This distorted payload is inserted into cover image to generate stego-image.

2.2.2 Embedding Algorithm

The embedding algorithm uses LSB Matching [11] and Matrix Encoding [12] to embed the distorted payload into the cover Image I. After inserting the distorted payload into the image the generated stego-image will be sent to the receiver for extraction.

Algorithm:

Step 1: Find the edge pixels in the cover image I using an edge detection algorithm.

Step 2: Convert the payload to its binary equivalent. Let this be denoted by B.

Step 3: Calculate the length of the payload. Let it be denoted by L.

Step 4: Calculate the number of pixels needed for embedding. Let this be denoted by pixnum. Set pixnum:=L/2.

Step 5: Set a counter i. Set K=1.

For i:=1 to pixnum
1. Set pix:=ith pixel values from the set of pixels selected in step 1.
2. Let p1=Red Plane LSB of pix, p2=Green Plane LSB of pix, p3= Blue Plane LSB of pix.
3. Let x1=B(K), x2=B(K+1).
4. Perform Embedding using checks for conditions mentioned in Table 1. Replace the original pixel components with the changed component values wherever necessary.
5. Set K=K+2.

End For

Step 6: Write the modified image matrix to a file.

Input: Cover Image I, Scrambled payload T

Output: Stego-Image.

The above algorithm finds the Edges by using the edge detection algorithm. Then the given payload is converted into binary equivalents denoted by B. The algorithm calculates the length of the payload denoted by L. The algorithm calculates the number of pixels needed for embedding denoted by pixnum. Then the algorithm searches the edges for embedding the data into the RGB pixels of edges until the payload is completely embedded.

2.3 PHASE II

The phase II is also called as extraction phase because in this phase II we are extracting the distorted payload from the stego-image that is generated during the phase I.

The extraction algorithm searches the edge regions of stego-image and extracts the hidden message using the decoding information. The cat map iteration information rem generated during phase I is used as key for this extraction algorithm.

Algorithm:

Step 1: Calculate number of pixels to search for hidden data as pixnum:=L/2.

Step 2: Set a counter i

.

For i:=1 to pixnum
1. Set pix:=i^{th} pixel values from the set of pixels supplied as decoding information.
2. Let p1=Red Plane LSB of pix, p2=Green Plane LSB of pix, p3= Blue Plane LSB of pix.
3. Perform XOR operation on p1 and p3 to get the first message bit x1. Perform XOR operation on p2 and p3 to get the second message bit x2 hidden in the pixel.
4. Store the message bits as a binary sequence.

End For

Step 3: Modify the sequence containing the binary message bits according to the appropriate type of the intended data. Let it be D.

Step 4: Perform descrambling of D using rem generated in Phase I as the key and Cat Mapping to get the original message.

Input: Stego-Image, Message Length L, Edge pixel information, descrambling key rem.

Output: Hidden Message.

The extraction algorithm calculates the number of pixels to search the hidden data. The pixel values are gated from the decoding algorithm. Let p1, p2 and p3 be the Red, Green and Blue pixels. Perform the XOR operations on p1and p3 to get the first message bits x1 then perform XOR operation on p2and p3 bits to get the second message bit x2 to get the hidden message pixels. Apply the procedure is applied until the completion of number of pixels. The generated bits are stored in the binary sequence.

The generated binary bits contain the scrambled payload. Descrambling is applied to the scrambled payload by using rem that is generated during embedding algorithm. Cat

mapping is applied to the binary bits to get the original message or payload.

2.4 TIME COMPLEXITY AND SPACE COMPLEXITY

Time complexity and space complexity of the proposed method is calculated. Let the number of pixels in the cover image be n. In proposed method first step is to find the edges in the cover image by using edge detection algorithm. The time required to find the edges will increase with the increase in the number of pixels in the cover image. So the time complexity to find the edges is O(n).

In the second step embedding algorithm is used to embed the payload. The embedding loop iterates pixnum times where pixnum=L/2 and L is the length of the payload. L<<n so the time complexity of the embedding algorithm is O(n).

Phase II time complexity is determined in the same manner. The de embedding algorithm loop iterates i from 0 to pixnum which is equal to L/2. The XOR operations is performed until the i value reaches to pixnum. Using rem value the original payload is generated by iterating the scrambled payload by iterating rem times. The time complexity of the de embedding algorithm is calculated as O(n).

Space complexity of the proposed steganography technique data structure sizes varies with the change in input is taken into consideration. Matrixes are used to store the cover image, stego-image and the secret message or payload. If n is the number of pixels in the cover image then the memory space required to store for n increases. Thus, the space complexity of the proposed mechanism is O(n).

3. EXPERIMENTAL WORK

The proposed mechanism is implemented by using java programming language. The proposed system login window is shown in figure 1. The figure 1 will show the login window where the authorized person will login into the system by using user id and password. The login user id and password is correct then it will show the embedded–de embedded window.

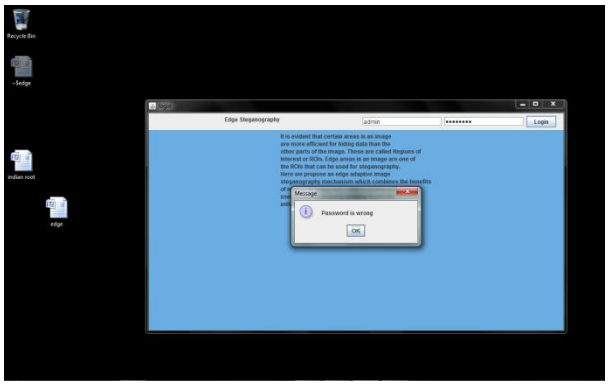

Figure 1: Login GUI

The embedded and de embedded window is shown in figure 2. The embedding and de embedding window consist of a text field, choose file button, rotate button, embed data, edge data and embedded data.

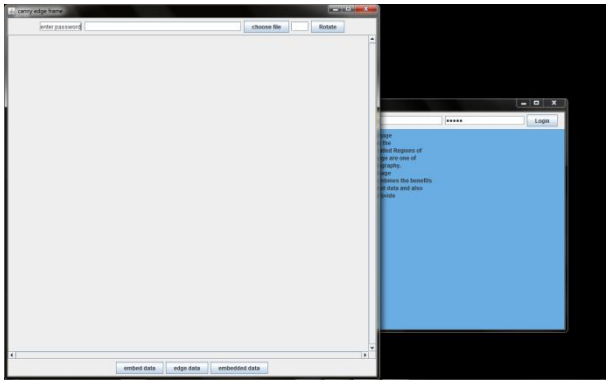

Figure 2: Embedded-De embedded GUI

The cover image selection window is shown in figure 3. When the choose file button is pressed the selection window will be opened in this window the cover image is selected either jpg or png image file.

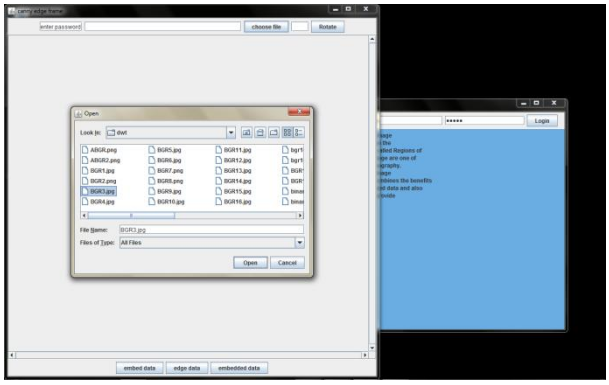

Figure 3: Cover-image selection GUI

The cover-image selection window will open the selected window. The selected image is shown in the figure 4. Then the edges are found by using the button edge data. The payload is inserted into the text field present in the window and no of rotations to be made is also given for the payload. After inserting the payload no of rotations press the button embed data then automatically the scrambled payload is inserted in to the image. The canny's edge detection algorithm, cat mapping, matrix encoding and LSB matching mechanisms are applied to embed the data into image.

Figure 4: embedding GUI

The edge detection GUI is shown in figure 5. The button edge data is used to generate the edge detected image. This button listener will implement the canny edge detection algorithm to generate edge detection GUI.

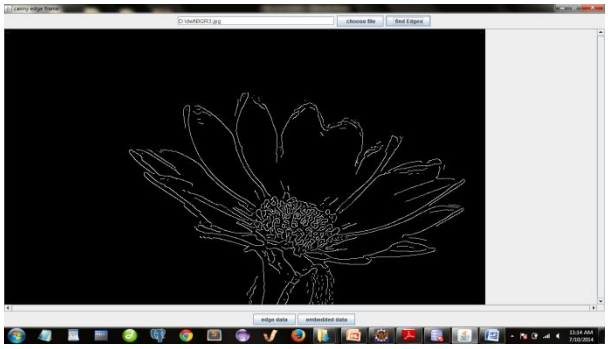

Figure 5: Edge Detection GUI

The de embedding window is shown in figure 6. After embedding the data into the image then the de-embedded button is used to get the scrambled payload. The scrambled payload is displayed in the small window.

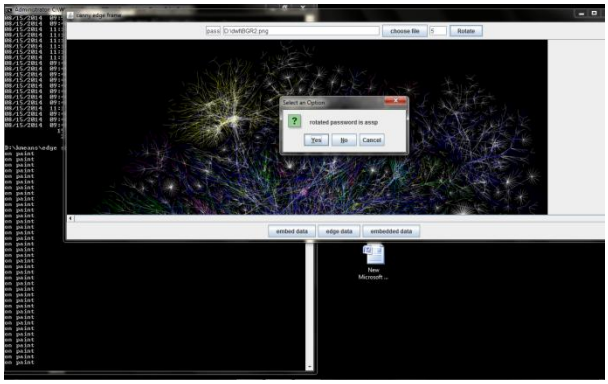

Figure 6: Embedded-De embedded GUI

The small window contains yes button. This button is used to get the original payload. The yes button performs the cat mapping algorithm to find the original payload. This button gives the original payload this is shown in figure 7.

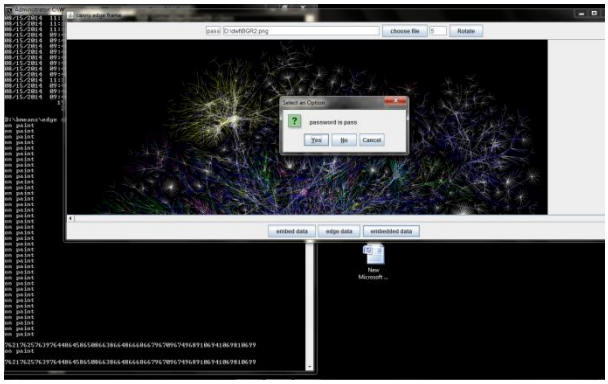

Figure 7: Payload generated GUI

4. RESULT ANALYSIS

The result analysis is done by calculating the MSE mean square error and PSNR peak signal to noise ratio. The distortion produced while embedding the payload in to the image is measured by using PSNR peak signal to noise ratio. The PANR is calculated by using MSE mean square error. The PSNR and MSE equations are given below.

$$MSE = \frac{1}{MXN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(x_{ij}-y_{ij}\right)^2 \qquad \textbf{(3)}$$

$$PSNR = 10\log_{10}\left|\left(\frac{255^2}{MSE}\right)dB\right.$$

(4)

In the above equation 4 M, N is the horizontal and vertical pixels dimensions of the cover image. x_{ij} and y_{ij} are the pixel values of the cover image.

Each and every RGB component in a pixel will be represented in 8 bits format. So each and every RGB pixel components maximum value is 255. In the equation 5 the 255 value represents the maximum color component of RGB pixel values. The 24-bit RGB images each color component has a color depth of 8 bits.

The cover image horizontal and vertical pixel dimensions are taken to calculate the MSE. First the MSE is calculated for each color plane then the average MSE is calculated by using individual MSE values. By using the MSE value the PSNR value is calculated.

Higher PSNR values indicate better fidelity of the stego image that indicates the low distortion. If the PSNR value is greater than 40 dB means human visual system cannot identify the distortion made to the cover image.

The proposed mechanism tests three images and calculates the PSNR values for the three images that will show that the proposed mechanism will produce less distortion. The proposed mechanism calculates the PSNR values by calculating the MSE values. The proposed mechanism will test the images with different pixel sizes such as 32X32, 60x60, 64X64, 80X80 and 100X100.

TABLE 2: The below table shows the PSNR values

Distortion measure for various levels of embedding			
PSNR(dB)			
Payload size	flower	fruit	Fountain
32X32	76.02	74.65	77.04
60X60	74.21	72.33	76.28
64X64	72.12	71.23	74.43
80X80	71.86	69.43	72.34
100X100	69.45	67.32	70.67
Average	72.623		

The result in Table 2 indicates that the proposed mechanism is having high PSNR peak signal noise ratio values. The average PSNR value for the proposed mechanism is 72.62 which is greater than 40 dB so that human visual system cannot identifies the change in the stego-image. The cover image is distortion is very low when compared to the other techniques. The proposed mechanism produces high fidelity.

4.1 χ2- steganalysis

The proposed mechanism will undergoes to the anti steganalysis test to check the security of the payload. The stego-image generated after embedding the payload into the cover-image is submitted to the anti steganalysis test. Here the proposed system will undergo to the χ2 steganalysis test to check the degree of imperceptibility of the proposed mechanism. The highest probability of existence of payload is detected as $6.346X10^{-4}$ shown in below figure 8.

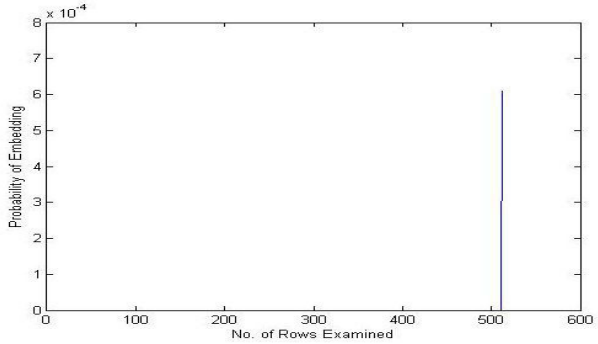

Figure 8: χ2 steganalysis test.

5. CONCLUSION

The Enhanced chaos based image steganography using edge adaptive and cat mapping is proposed to enhance the addition security to the payload while embedding and finds Edges in the images which are good Regions of Interests (ROIs) for hiding the payload in to the image by using canny edge detection algorithm. The proposed system uses cat mapping transformation to provide the security to the payload. The proposed mechanism uses matrix encoding and LSBM least significant bit matching to embed the scrambled payload in to the image. The proposed technique exhibits high fidelity and good imperceptibility and with stands χ2 attack.

Future extinction will focus on possible reduction and extends the mechanism for higher order image planes.

6. REFERENCES

[1] John Canny, "A computational approach to edge detection", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8, No. 6, pp.679–698, Nov. 1986.

[2] Zhu Liehuang, Li Wenzhuo, Liao Lejian , Li Hong, "A Novel Algorithm for Scrambling Digital Image Based on Cat Chaotic Mapping", International Conference on Intelligent Information Hiding and Multimedia Signal Processing, pp. 601-605, 2006.

[3] Ron Crandall, "Some Notes on Steganography", Posted on Steganography Mailing List, 1998. Source: http://www.dia.unisa.it/~ads/corsosecurity/ www/CORSO-0203/steganografia/LINKS%20LOCALI/matrix-encoding.pdf

[4] Weiqi Luo, Fangjun Huang, Jiwu Huang, "Edge Adaptive Image Steganography Based on LSB Matching Revisited", IEEE Transactions on Information Forensics and Security, Vol. 5, No. 2, June 2010, pp. 201-214.

[5] Grant Kelly, Bruce McKenzie, "Security, privacy and confidentiality issues on the internet", Source: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1761937.

[6] C.V. Serdean, M. Tomlinson, J. Wade, A.M. Ambroze, "Protecting Intellectual Rights: Digital Watermarking in the wavelet domain", IEEE Int. Workshop Trends and Recent Achievements in IT, pp. 16-18, 2002.

[7] Ratnakirti Roya,*, Anirban Sarkara, Suvamoy Changdera, "Chaos based Edge Adaptive Image Steganography", International Conference on Computational Intelligence: Modeling Techniques and Applications (CIMTA) 2013

[8] G.Karthigai Seivi, Leon Mariadhasan, K. L. Shunmuganathan, "Steganography using Edge Adaptive Image", Proc. of the International Conference on Computing, Electronics and Electrical Technologies (ICCEET), pp. 1023 1027, 2012.

[9] Cheng-Hsing Yang, Chi-Yao Weng, Shiuh-Jeng Wang , Hung-Min Sun, "Adaptive Data Hiding in Edge Areas of Images With Spatial LSB Domain Systems", IEEE Transactions on Information Forensics and Security, Vol. 3, No. 3, September 2008, pp.488-497.

[10] Qian-chuan Zhong, Qing-xin Zhu , Ping-Li Zhang , "A Spatial Domain Color Watermarking Scheme based on Chaos", International Conference on Apperceiving Computing and Intelligence Analysis (ICACIA), pp. 137-142, 2008.

[11] X. Li, B. Yang, D. Cheng, and T. Zeng, "A generalization of lsb matching", IEEE Signal Processing Letters, vol. 16, no. 2, pp. 69-72, 2009.

7. AUTHORS BIOGRAPHY

V. Lokeswara Reddy did his M.Tech (CSE) from SRM University, Chennai in the year 2005. He did his M.C.A from S.V. University, Tirupati in the year 2000. He is pursuing his Ph.D from JNTUA, Anantapur. He has a total of 13 years of experience in teaching. Currently he is working as Associate Professor at K.S.R.M College of Engineering, Kadapa. He has presented 9 papers in International, National Conferences and published 9 papers in International journals.

B. Sailendar did his B.Tech (IT) from JNTUA studied in AITS in the year 2011. He is pursuing his M.Tech (CSE) from KSRM College of Engineering affiliated to JNTUA Anantapur University in the year 2014.