Information Security and Computer Fraud, 2015, Vol. 3, No. 1, 8-14
Available online at http://pubs.sciepub.com/iscf/3/1/2

DOI:10.12691/iscf-3-1-2

Novel Chaos Based Steganography for Images Using Matrix Encoding and Cat Mapping Techniques

V. Lokeswara Reddy*

Department of CSE K.S.R.M College of Engineering Kadapa, Y.S.R. District., A.P., India
*Corresponding author: VL_REDDY@YAHOO.COM

Received July 04, 2015; Revised July 15, 2015; Accepted July 22, 2015

Abstract Steganography is an ancient technique of data hiding. Steganography is a technique in which secret data is hidden into vessel image without any suspicion. Most of the traditional approaches are achieving limited data hiding. To expand the information covering up proposed mechanism uses various techniques. Proposed mechanism is increasing the info disappearing ability and also provides high imperceptibility towards steganalysis. The key goal connected with steganography is usually providing the protection for hidden information from adversary. The integrity and confidentiality of the electronic information offers occur underneath immense hazard. Steganography will be the art along with technology associated with coated composing. Steganography utilizes text, pictures, audio tracks, video etc. as being a protect moderate. The proposed mechanism is based on image steganography technique. It is implemented based on the Canny Edge Detection. Pixel positions of edges in conjunction with Chaotic Cat Mapping [10] are used for the steganography technique. Proposed mechanism is using edge adaptive image steganography [3] by combining the benefits of matrix encoding and LSBM [8,9].

Keywords: *payload, chaotic cat mapping, matrix encoding, LSB Matching, Regions of Interests (ROI), edge detection, image steganography*

Cite This Article: V. Lokeswara Reddy, "Novel Chaos Based Steganography for Images Using Matrix Encoding and Cat Mapping Techniques." *Information Security and Computer Fraud*, vol. 3, no. 1 (2015): 8-14. doi: 10.12691/iscf-3-1-2.

1. Introduction

Steganography is the art of hiding the message in a medium so that it is not visible to the intruders. Steganography is providing security through obscurity. Steganography mechanism is used in ancient days. The word steganography is born from the Greek word Stefano's which means secret or covered and graphe means writing are drawing. In order to hide the messages steganography is having different mediums where we can hide the message they are text files, audio files, video files, image files etc.

The main target involving steganography is providing security towards message so that intruders are not able to detect the messages. Steganography concealed the particular message and so there is zero familiarity with the actual existence of the message. To hide the message steganography chooses the medium generally calling as cover. The cover may be image file, audio file, video file or even text file. Steganography allows many techniques to insert the message in to the cover. Steganography allows mechanisms to insert the payload into the cover such as LSB mechanism MSB mechanism etc. after hiding the payload into the cover the covers are send across the Network. The destination will receive the cover and performing reverse mechanism to find the payload. Generally steganography will be used with cryptography to provide more stability to the payload..

There are four ways to implement steganography.

1. Using text files.
2. Using images.
3. Using audio files.
4. Using video files.

Steganography uses any of the above mediums. The message first changed into binary format then by using the techniques such as LSB, MSB, 3-3-2 LSB, LSBM mechanisms the message will be hidden in to the medium.

2. Proposed Mechanism

Embedding payload in to distinct parts of fascination in the photograph is usually a reasonably fresh procedure for image steganography. ROIs could also be any object within the image that produces slightest of distortion once embedded with knowledge.

One such ROI is the edge areas in a picture. Edge locales are suitable for information concealing on the grounds that the human visual framework is less touchy to contortions in edge districts and it additionally gives randomized pixel positions. Installing into randomized pixel positions disperses the payload all through the spread and decreases the likelihood of location by steganalysis analyzers. ROI based steganography can be thought to be a suitable option for advancement of better steganography algorithms. On the other hand, there are varied approaches to ensure the security of the payload. Normal methodologies incorporate pre-encrypting the payload or misshaping it in a pseudo arbitrary way utilizing different mathematical transformations and mappings. This approach means to present a ROI based spatial area steganography instrument that inserts payload information into edge areas of a spread picture.

With a specific end goal to improve the security of the payload it is subjected to a pre-embedding bending utilizing a chaotic mapping system which guarantees that the real message does not get uncovered even on exposure to an intruder [4] having information about the implanting component.

The proposed mechanism works on the spatial domain and uses Canny's Edge Detection [1] spotting the edge pixels in the spread picture and shroud information into the chosen pixels. Canny's method has high immunity to noise and can detect true weak edges. It has been considered to be an optimized and standard method for detecting edges in images as compared to other techniques of edge detection [6]. Edge versatile implanting gives vital pixel randomization as edge areas are scattered pretty much all through the spread picture. With an end goal to expand the security of the payload even on account of presentation to an adversary, Cat Mapping [2] is connected to mutilate the payload at first.

An interesting feature of Cat mapping is that if it is applied to an image, after a certain number of iterations [2] the mapping returns the original image. The middle stages are however contorted and look to some extent like the first picture. It is an area preserving map and the basic operation of the map is that the picture is sheared one unit up, then one unit to one side, and all that lies outside that unit square is moved back by the unit until it's inside the square.

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 stages:

Step 1: Smoothing.

- Obscuring of the picture to evacuate noise.

Step 2: Finding Gradients.

- The edges ought to be checked where the slopes of the picture has large magnitudes.

Step 3: Non-Maximum Suppression.

- Just neighborhood maxima ought to be checked as edges.

Step 4: Double Thresholding.

- Potential edges are determined by thresholding.

Step 5: Edge Tracing by Hysteresis.

- Last edges are dictated by smothering all edges that are not associated with an exceptionally certain (strong) edge.

Input: Cover Image.

Output: Edges founded in Image

Canny edge detection algorithm is used to find the edges of an image. Canny edge detection algorithm is efficient in finding the strong edges. Canny edge detection algorithm is applied to the image in 5 steps. After applying the steps we will get the image in which the edges are detected. After finding the edges the next step in this module is to insert the distorted payload in to the image for inserting the payload the below algorithm is used.

To increase the security to the payload the additional layer of security is provided by applying the chaotic cat mapping [9] 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.

$$M\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \mod 1 \tag{1}$$

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

$$M\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & p1 \\ p2 & p1p2 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} \mod N \tag{2}$$

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 quantity of emphases needed to get the first payload. The second step is performed at the time of extracting the payload from the stego-image.

The embedding is performed utilizing a mix of LSB Matching and Matrix Encoding [20]. Matrix Encoding guarantees that the payload is installed into the cover picture 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 [7]. 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

$$x1 = p1 \oplus p3 \tag{3}$$

$$x2 = p2 \oplus p3 \tag{4}$$

All necessary actions that need to be taken during embedding.

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.

Table 1. Action List for Embedded Conditions

Possibilities	Appropriate Action to be taken
x1= p1 ⊕ p3 x2= p2 ⊕ p3	No change Required
x1≠p1 ⊕p3 x2= p2 ⊕p3	Change component R to match condition(3) & (4)
x1=p1 ⊕ p3 x2≠p2 ⊕ p3	Change component G to match condition(3) & (4)
x1≠p1 ⊕p3 x2≠ p2 ⊕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. Encoding Phase

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: Discover the quantity of Cat map transformations needed to distort the payload and recover the first once more. it is signified 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). It is 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

By using this algorithm the original payload is scrambled to enhance the security.

This algorithm discovers the quantity of cat map transformations needed to distort the payload and regenerate the payload [5]. P will represent 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 [10] and Matrix Encoding 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: Discover the edge pixels in the cover image I utilizing 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. It is denoted by L.

Step 4: Figure the quantity of pixels required for inserting. Let this be signified 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 arrangement of pixels chosen 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. Supplant the original pixel segments with the changed segment values wherever fundamental.
5. Set K=K+2.

End For

Step 6: Compose the modified image to a document (or) 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 calculation ascertains the length of the payload indicated by L. The calculation computes the quantity of pixels required for installing meant by pixnum. Then the algorithm searches the edges for embedding the data into the RGB pixels of edges until the payload is completely embedded. Here there is chance to apply image encryption algorithm [11].

2.3. Decoding Phase

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: Compute number of pixels to look for concealed information as pixnum:=L/2.

Step 2: Set a counter i

For i:=1 to pixnum

1. Set pix:=ith pixel values from the arrangement of pixels supplied as disentangling data.
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. Space Complexity and Time Complexity

Space multifaceted nature of the proposed steganography method 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. In the event that n is the quantity of pixels in the spread picture then the memory space needed to store for n increments. Thus, the space complexity of the implemented mechanism is O(n).

3. Experimental Work

The proposed mechanism is implemented by using one of the java programming language element called swings. The proposed mechanism is provided with login security and password. The first window is login window which will provide the user authentication system. In the event that the client is not verified the proposed framework is not permitted to utilize this instrument. The Login GUI window is shown in Figure 1.

Figure 1. Login GUI

In the event that the client is confirmed then the following window is opened that is installing and de-inserting window.

The embedding and de embedding 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.

The embedding and de-embedding window consists of the text field. In this text field whatever the message or payload we want to hide that will be typed into that corresponding text field. After the payload is inserted then the image selection button will be clicked i.e choose file. At the point when the button is clicked the cover image selection window will be opened. Then the image is selected in which the payload wants to be hiding. The image selection window is Figure 3.

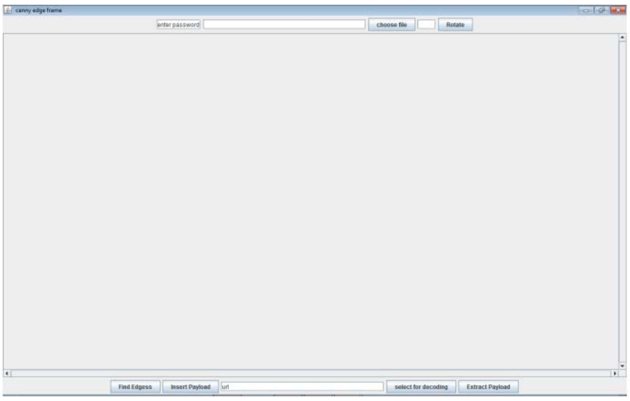


Figure 2. Embedded-De embedded GUI

The cover image determination window is demonstrated 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.

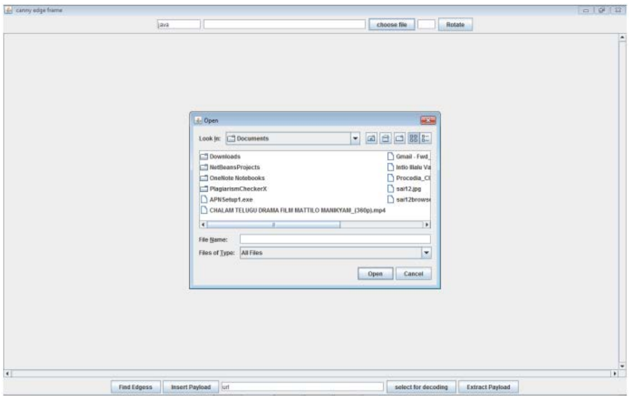


Figure 3. Cover-image selection GUI

Then the cover image is displayed on the window that is shown in Figure 4.

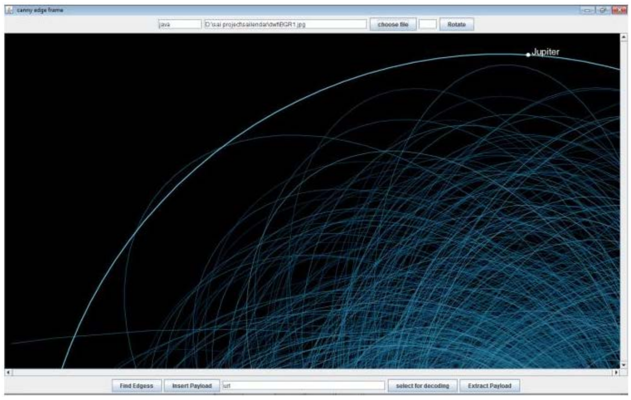


Figure 4. Displaying the Cover Image GUI

The next step is to enter the number of rotations that need to be made by the cat mapping algorithm. After entering the number of rotations then the rotate button needed to be clicked. After clicking on the rotate button the cat mapping transformation algorithm scrambles the payload and the mutilated payload will be shown by utilizing the little window. The window is shown in the Figure 5.

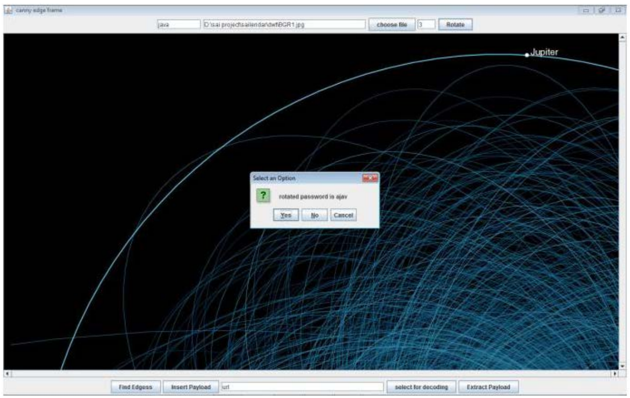


Figure 5. Distorted Payload Displaying GUI

If the distorted payload is correct then the user will click the yes button. After clicking the yes button then within the same Window the user will click on the find edges button. The find edges button will invoke the action listener and calls the canny edge detection algorithm. The canny edge detection algorithm will find the edges based on the scrambled payload and display the image. The corresponding widow is shown in Figure 6.

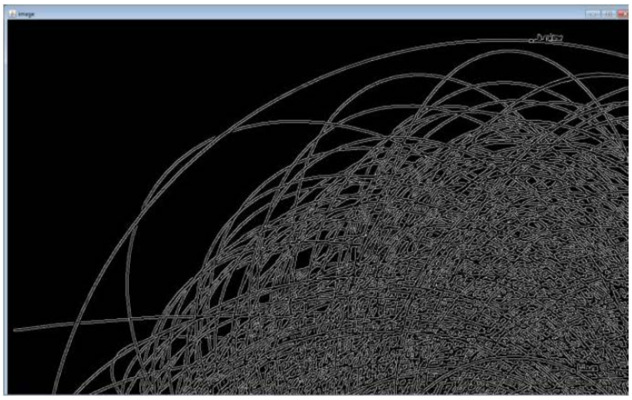


Figure 6. Canny Edge Detection Image

After finding the edges the scrambled payload is inserted into the image by clicking on the button Insert Payload. At whatever point the client is clicked on the supplement payload catch the payload is embedded into the picture and a picture determination window is opened to spare the created picture. The generated image is saved with some name into the storage device. The image selection window is shown in Figure 7.

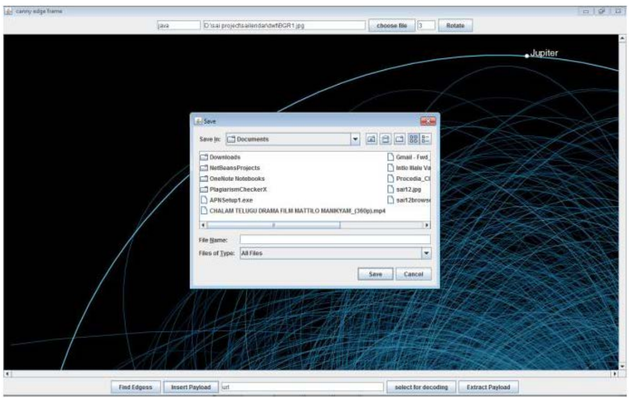


Figure 7. Image selection window for saving image GUI

Then after saving the image in to storage device the image is sended to the destination. After receiving the image at destination again the image is selected for de-embedding the payload which is hidden in the image.

The select for decoding button is used to select the image for decoding. When the user clicked on this button the window selection screen is opened. The Window is shown in Figure 8.

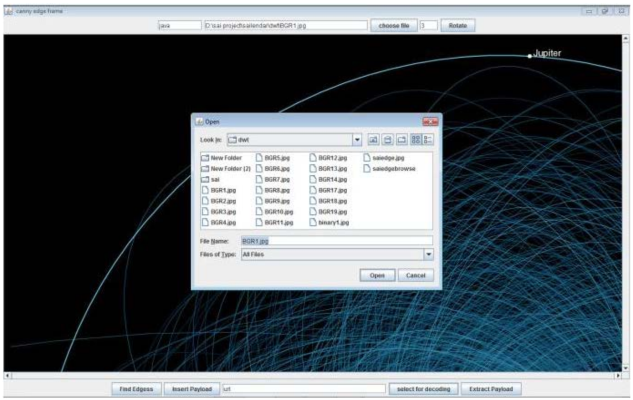


Figure 8. Stego Image Selection GUI

Then the user at the destination side will select the image from storage device. After clicking on the open button the image is displayed on the screen the window is shown in the Figure 9.

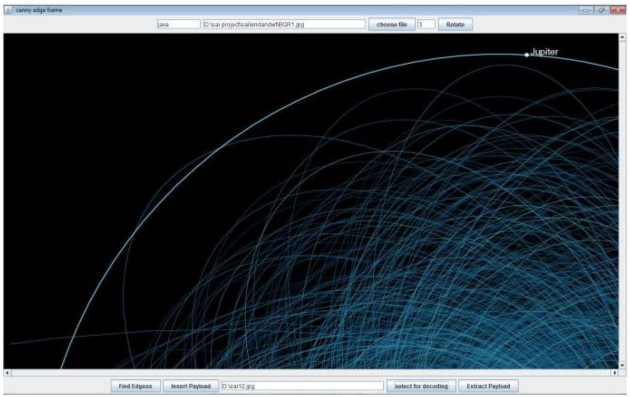


Figure 9. Stego Image Displaying Screen GUI

The Stego-Image is taken as the information for the vigilant edge discovery calculation and the edges are recognized in light of the bended payload.

After finding the edges by using edge detection algorithm then 3-3-2 LSB mechanism is applied to find the scrambled payload across the image.

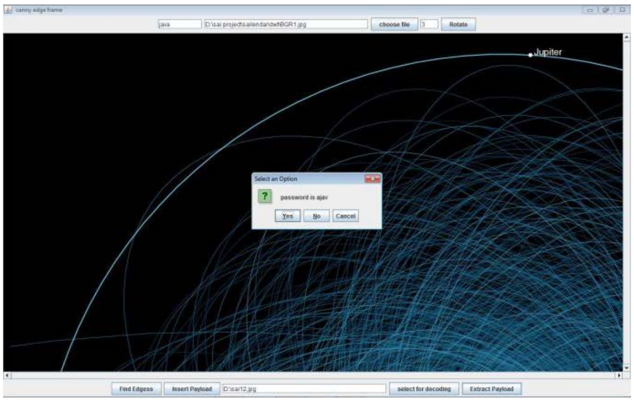


Figure 10. Scrambled Payload Displaying GUI

The 3-3-2 LSB mechanism is applied after finding the edges. The edges consist of RGB Pixel values. Here RGB stands for Red Green Blue colors. Form Red component the last 3 bits are taken and from the Green component the last 3 components will be taken and from the Blue component the last 2 bits are taken and combined the 3,3,2 bits into one byte and that byte value is taken to find the corresponding character to display. In the wake of discovering all the characters the contorted payload is shown in the little window indicated in Figure 10.

If the user clicks the yes button in the window at that point the feline mapping calculation discovers the quantity of pivot required to find out the original payload and finds the original payload displays into the window. The window is shown in the Figure 11.

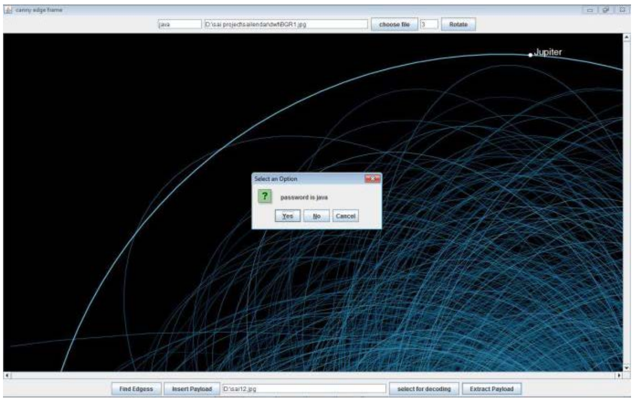


Figure 11. Original Payload Displaying GUI

4. RESULT ANALYSIS

performance estimation for picture bending is extraordinary Peak signal to noise ratio(PSNR) which is characterized under the distinction contortion measurements and can be connected on stego pictures. PSNR is used to evaluate quality of stego image after embedding the secret message. Secret message can be any word. The execution as far as limit and PSNR (in dB) is shown for the system in the accompanying subsections. PSNR is defined as per Eq.5

$$MSN = \frac{1}{MXN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(a_{ij} - b_{ij}\right)^2 \qquad (5)$$

$$PSNR = 10\log_{10}\left|\left(\frac{255^2}{MSN}\right)\right| dB \qquad (6)$$

Here aij and bij speaks to pixel estimations of unique spread picture and stego picture separately as in Eq.6 The figured PSNR as in Eq.6 ordinarily receives dB esteem for quality judgment, the bigger PSNR is higher the picture quality (which implies there is a little contrast between spread picture and stego picture). Actually littler dB quality means there is a more mutilation. PSNR qualities falling underneath 30dB demonstrate reasonably a low quality. However, high quality strives for 40dB or more.

Result Discussion of proposed work after embedding secret data in cropped image, resulted cropped stego image is shown in Figure 9. After performing decoding process on stego image, retrieved output of the secret message is shown in Figure 11.

The outcome examination is carried out by computing the PSNR peak signal to noise ratio and MSE mean square error. 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 in Eq.5 & Eq.6.

In the above comparison 3 M, N is the level and vertical pixels measurements of the spread picture. aij and bij are the pixel estimations of the spread picture.

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.

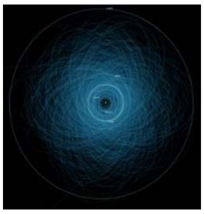
The cover image horizontal and vertical pixel dimensions are taken to calculate the MSE. First the MSE is calculated for each color plane then the average MSE is calculated by using individual MSE values. By using the MSE value the PSNR value is calculated.

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 2a. The below table shows the PSNR values

IMAGINATION

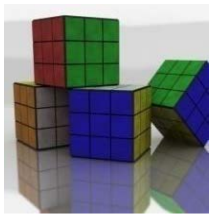


JUPITER

FLOWER

MUSIC

MONEY



CUBE

Table 2b. Measuring the distortion of various levels of embedding

Distortion measure for various levels of embedding					
Payload size PSNR(dB)					
Name of the Image	32X32	60X60	64X64	80X80	100X100
Imagination	76.02	74.65	74.23	77.04	76.12
Jupiter	74.21	72.33	69.32	76.28	72.34
Flower	72.12	71.23	70.98	74.43	67.23
Music	71.86	69.43	67.23	72.34	74.55
money	69.45	67.32	75.34	70.67	71.23
cube	68.34	72.13	70.12	71.31	74.34
Average	72.62				

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 were for than 40 dB so that human natural visual system cannot identifies the change in the stego-image. The cover image is distortion is very low when contrast with the other techniques. The proposed mechanism produces high fidelity.

5. Conclusion

The Enhanced chaos based image steganography utilizing edge adaptive and feline cat mapping is proposed to improve the expansion security to the payload while installing and discovers Edges in the pictures which are great Regions of Interests (ROIs) for concealing the payload into the picture by utilizing vigilant edge discovery calculation. The proposed system uses cat mapping transformation to provide the stability and 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 $\chi 2$ attack.

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

References

[1] John Shrewd, "A computational way to deal with edge identification", IEEE Exchanges on pattern analysis and Machine Intelligence, Vol. 8, No. 6, pp.679-698, Nov. 198.

[2] 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

[3] 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.

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

[5] 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.

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

[7] 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.

[8] 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.

[9] R. L. Tataru, D. Battikh, S. El Assad, H. Noura, O. Deforges, "Enhanced Adaptive Data Hiding in Spatial LSB Domain by using Chaotic Sequences", Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, pp. 85-88, 2012.

[10] Sahar Mazloom, Amir-Masud Eftekhari-Moghadam, "Color Image Cryptosystem using Chaotic Maps", IEEE Symposium on Computational Intelligence for Multimedia, Signal and Vision Processing, pp. 142-147, 2011.

[11] Chen Wei-bin, Zhang Xin, "Image Encryption Algorithm based on Henon Chaotic System", International Conference on Image Analysis and Signal Processing (IASP), pp. 94-97, 2009.