INTERNATIONAL JOURNAL OF RECENT TRENDS IN ENGINEERING & RESEARCH
Impact Factor 4.101 ISSN -2455-1457
RECENT ADVANCES IN ELECTRONICS & COMMUNICATION ENGINEERING (RAECE-2K17)

A remarkable Watermarking technique for PNG image file formats using Even Odd concept

Ms. Ch. Bharathi
M.Tech Pursuing, Dept. of ECE
NNRG, Hyderabad, India

***Abstract* — The quick increment in the high limit computerized recording gadgets and the web innovation is turning into a risk to the media content proprietors. This progression influenced unlawful clients to share to or disseminate the substance like picture, sound, video and so on. effortlessly and illegally. Advanced watermarking on media is a rising exploration to give better security and genuineness. The information embedding plan modifies the cover substance, for example, picture, sound, content and so on., in information stowing away. The information to be embedded can be mystery content, personality of the substance proprietor and so forth., contingent on the application for which embedded plan is designed.**

This paper presents a reversible watermarking technique based on even odd concept. Only one of the RGB components in pixel is changed during this embedding process. This data embedding technique gives balance between better visual quality, simple algorithm and higher data hiding capacity. This is done by transforming any image file into PNG file format. The experimental results show PSNR values of the watermarked image above than the satisfactory level.

Keywords* — *Reversible watermarking, data embedding, PSNR(Peak Signal to Noise Ratio), Even-Odd

I. INTRODUCTION

The process of hiding information in other multimedia content like image, audio, video or text originated many years ago. In the last two decades much amount of research has been done. Steganography or Digital watermarking provides copyright information, broadcast monitoring, source tracking etc., The process which enables images to be authenticated and then restored to their original form and extracting the secret data as it is embedded. This is known as Reversible Watermarking which makes the images acceptable for legal purposes. In this process the cover content and the data embedded can be exactly restored. In medical and military applications Information is very sensitive. Small change in the pixel values or hidden data is not acceptable.

The image format also plays a major role while embedding the data. PNG (portable Network Graphics) is one of the good lossless format specifically designed for the web. It is saved with a maximum of 256 colors which saves more efficiently. PNG can support 24 bit color images and also grayscale images.

There are many techniques for reversible watermarking like Histogram Shifting [1] where histogram modification is done to allow good space for data embedding. There are some algorithms which can consider Overflow and underflow problems that occurs because of histogram shifting. The technique Difference Expansion [2] gives a wonderful technique where difference of the neighboring pixels and average integer values are considered. Some mathematical proceedings are done to embed the data and inversely to extract. Even-Odd concept [3], this technique includes generation of blueprint for original image in the embedding phase and this blueprint is used to extract the embedded data and cover image maintaining good PSNR values. Prediction Error Expansion [4], this technique aimed to reduce embedding distortion. To split the expanded difference between current pixel value and its predicted context.

Embedding is done by inserting the expanded difference into current pixel. Inversely, expanded error is recovered, data is extracted and image is restored. All these schemes could not maintain balance between Data embedding capacity, Robustness, Visual quality and Algorithm complexity. In this proposal the method alters a small component in the pixel value so good PSNR values can be achieved by using simple algorithm.

II. PROPOSED SCHEME

The proposed technique is to embed the data in to an image by using even odd concept which embeds one secret bit into one pixel. Firstly a grayscale image of any format should be changed to PNG. Now the grayscale image which is in PNG format shows the Red, Green and Blue values equal (which is our finding). The formula for data embedding is different according to pixel values either even or odd and also secret bit. A watermarked image is constructed after coding into the cover image and inversely the watermarked image is used to extract the secret data and the cover image.

A. Algorithm for Watermarking/ Data Embedding:

Input: Cover image and Secret data
Output: Watermarked image
Let I and S are the original image and secret bits respectively of the same size m*n. 1 denotes Red matrix, 2 denotes Green matrix and 3 for Blue.

Step 1: Secret data of size equal to image size is generated
Step 2: Data embedding

1: if I(i,j,1) is even & secret bit is 1
2: I(i,j,1)is incremented by 1
3: elseif I(i,j,3) is odd & secret bit is 0
4: I(i,j,3) is incremented by 1
5: elseif I(i,j,1) is 255 & secret bit is 0
6: I(i,j,3)decremented by 1
7: end

Here i indicates the row value and j indicates the column value 1 means the Red color, 2 means Green and 3 means Blue color. In the algorithm first condition if one pixel value is even and secret bit is odd the red color representation will be incremented by 1 and the remaining two colors remains the same. In the second condition if the pixel value is odd and the secret bit is even the blue color representation will be incremented by 1 and remaining two colors remains the same. In the third condition if the pixel value is 255 and the secret bit is even the blue color will be decremented by 1 as there are no color representations beyond 255. Finally if the pixel value is odd and the secret bit is odd no changes are done and if the pixel values are even and the secret bit is even no changes are done. This forms the embedding process.

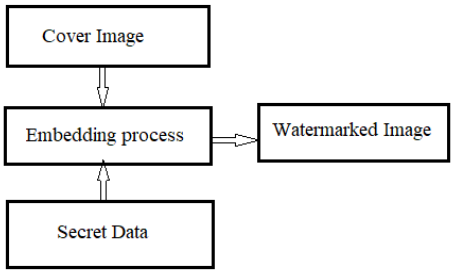

Fig. 1. Block diagram for generating Watermarked image

Let's look into an example for concept clarity. For instance:

Secret data S = $\begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$

Below are the pixel values of original image and image after data embedding is done

R:200 G:200 B:200	R:206 G:206 B:206	R:173 G:173 B:173
R:222 G:222 B:222	R:160 G:160 B:160	R: 92 G: 92 B: 92
R:141 G:141 B:141	R:104 G:104 B:104	R: 67 G: 67 B: 67

Fig 2: Pixel values of original image

R:200 G:200 B:200	R:207 G:206 B:206	R:173 G:173 B:173
R:222 G:222 B:222	R:160 G:160 B:160	R: 92 G: 92 B: 92
R:141 G:141 B:142	R:104 G:104 B:104	R: 67 G: 67 B: 68

Fig 3: Pixel values after embedding data

B. Algorithm for extracting cover image and secret data:

Input: Watermarked image
Output: Cover image and secret data

Step 1: Extraction of cover image

1: if E(i,j,1)is greater than E(i,j,2)
2: E(i,j,1)is decremented by 1
3: elseif E(i,j,3)greater than E(i,j,2)
4: E(i,j,3) is decremented by 1
5: elseif E(i,j,3) is less than E(i,j,2)
6: E(i,j,3)incremented by 1
7: end

In the embedding process we did not change the green color representation. So to extract the secret data we are comparing the Red color representation and Blue color representation with middle one which is Green color. Now for extracting the original image firstly if red color representation is greater than green it should be decremented and if blue color value is greater or lesser than green color value it should be decremented or incremented accordingly so that all the values in the pixel will be equal. This makes the image extraction.

Step 2: Extraction of secret data

1: if E(i,j,1),E(i,j,2),E(i,j,3) are equal & E(i,j,2) is even
2: S1(i,j)=0
3: elseif E(i,j,1),E(i,j,2),E(i,j,3) are equal & E(i,j,2) is odd
4: S1(i,j)=1
5: elseif E(i,j,1)greater than E(i,j,2) & E(i,j,2) is even
6: S1(i,j)=1
7: elseif E(i,j,3)greater than E(i,j,2) & E(i,j,2) is odd
8: S1(i,j)=0
9: elseif E(i,j,3)less than E(i,j,2) & E(i,j,2) is odd
10: S1(i,j)=0
11: end

For the extraction of the secret data we need to consider the embedded image and by comparing the values of Red, Green and Blue we can extract the secret data. If the Red, Green,

Blue color component values of the pixel are same and even the secret bit which is embedded is even that is 0. If the Red, Green, Blue color component values of the pixel are same and odd the secret bit which is embedded is odd that is 1. If the Green component is even and red component value is greater by 1 the embedded bit is 1. If the green component is odd and the blue component is greater by 1 then the embedded bit is 0. For 255 it is exceptional case. If blue component value is lesser by 1 the embedded bit is 0. In this way we can explain the code for extraction of secret data

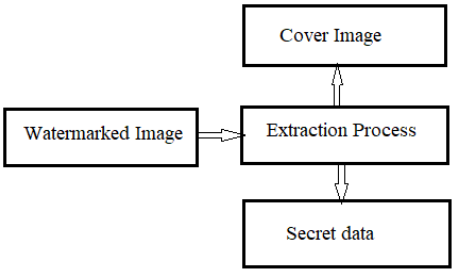

Fig 4: Block diagram for extracting cover image and secret data

R:200 G:200 B:200	R:206 G:206 B:206	R:173 G:173 B:173
R:222 G:222 B:222	R:160 G:160 B:160	R: 92 G: 92 B: 92
R:141 G:141 B:141	R:104 G:104 B:104	R: 67 G: 67 B: 67

Fig 5: Pixel values after extracting the data

Secret data after extraction S1= $\begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$

III. EXPERIMENTS AND RESULTS

1) An image when converted from any format to PNG the representation of pixel values are different. The single intensity value to RGB of same values.

Fig 6: Lena image in Grayscale TIF format

Fig 7: Lena image in Grayscale PNG format

Pixel Information and corresponding intensity values:
(112,124) (113,124) (114,124) (115,124)
(112,125) (113,125) (114,125) (115,125)
(112,126) (113,126) (114,126) (115,126)
(112,127) (113,127) (114,127) (115,127)

29	43	139	130
35	117	159	91
80	162	114	115
142	150	113	168

Fig 8: Pixel values of image in TIF

R: 29 G: 29 B: 29	R: 43 G: 43 B: 43	R:139 G:139 B:139	R:130 G:130 B:130
R: 35 G: 35 B: 35	R:117 G:117 B:117	R:159 G:159 B:159	R: 91 G: 91 B: 91
R: 80 G: 80 B: 80	R:162 G:162 B:162	R:114 G:114 B:114	R:115 G:115 B:115
R:142 G:142 B:142	R:150 G:150 B:150	R:113 G:113 B:113	R:168 G:168 B:168

Fig: 9 Pixel values of image in PNG

2) Finding the Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR):

The two mainly used error metrics to compare the difference between original image and the compressed image are MSE and PSNR. The MSE is used to calculate the cumulative difference error between two images. The squaring is necessary as to eliminate the negative signs. Whereas PSNR is used to find the error value.

The formulae for both metrics can be given as:

$$MSE = \left(\frac{1}{MN}\right)\sum_{x=1}^{M}\sum_{y=1}^{N}\left(I(x,y) - I'(x,y)\right)^2 \quad \ldots \text{Eq. (1)}$$

$$PSNR = 20 * log10\left(\frac{255}{sqrtMSE}\right) \quad \ldots \text{Eq. (2)}$$

Here I(x,y) is the original image, I' is the altered image, M,N are the dimensions of the image.

TABLE 1
Embedding capacity, MSE and PSNR values of the images

Picture name	***Size***	***Bits embedded***	***MSE***	***PSNR***
Cameraman	512x512	262144	0.08 0.00 0.08	58.91 Infinite 58.93
House	512x512	262144	0.06 0.00 0.11	60.24 Infinite 57.90
Lake	512x512	262144	0.08 0.00 0.08	58.91 Infinite 58.94
Lena	256x256	65536	0.08 0.00 0.08	58.94 Infinite 58.90
Peppers	512x512	262144	0.08 0.00 0.09	59.04 Infinite 58.83

3) Comparison of the secret data

If S(I,j) is equal to S1(I,j) we can call it as Success bit and if they are not equal we call it as Error bit. The result is:
Number of Success bits: 262144
Number of Error bits: 0

4) Comparison of Original Images and Embedded images:

We present different images in PNG format of size 512x512 and 256x256 and their corresponding watermarked images can be observed. Even though the difference cannot be seen with our naked eye we can spot them by using error metrics

IV. CONCLUSION

The proposed scheme of data hiding is simple concept based on Even Odd technique. The distortion in the image is very less compared to many other techniques. This maintains the balance between data hiding capacity and the visual quality of the picture. The experimental results also show better PSNR values. With this new invented technique many sophisticated schemes can be proposed.

REFERENCES:

[1] Sang-Kwang Lee, Hyang-Mi Yoo & Jae-Won Suh, "Reversible data hiding employing histogram shifting using a rotated even–odd difference image"

[2] Jun Tian, "Reversible Data Embedding Using a Difference Expansion", IEEE transactions on Circuits and Systems for Video technology, vol. 13, no. 8, August 2003

[3] Arijit Kumar Pal, Poulami Das, Nilanjan Dey, "Odd-Even Embedding Scheme Based Modified Reversible Watermarking Technique Using Blueprint"

[4] Dinu Coltuc, "Improved Embedding for Prediction-Based Reversible Watermarking", IEEE transactions on Information Forensics and Security, Vol. 6, no. 3, September 2011

[5] Sagar Gujjunoori*, B.B. Amberker, "DCT based reversible data embedding for MPEG-4 video using HVS characteristics", Elsevier Information security technical report

[6] J. Tian, "Wavelet-based reversible watermarking for authentication," in Security and Watermarking of Multimedia Contents IV—Proc. SPIE, E. J. Delp III and P. W. Wong,
Eds., Jan. 2002, vol. 4675, pp. 679–690.

