Pixel Authorized by Pixel to Trace with SFC on Image to Sabotage Data Mugger: A Comparative Study on PI Stego

Rengarajan Amirtharajan and John Bosco Balaguru Rayappan
Department of Electronics and Communication Engineering, School of Electrical and Electronics Engineering, SASTRA University, Tamil Nadu, 613401, India

Corresponding Author: Rengarajan Amirtharajan, Department of Electronics and Communication Engineering, School of Electrical and Electronics Engineering, SASTRA University, Tamil Nadu, 613401, India

ABSTRACT
There is a raging moral combat to secure information from nefarious attackers as infringement of data is alarmingly escalating with valuable secret information being sabotaged, manipulated or even sold. This has made it necessary for development of efficacious information hiding algorithm to prevent info-sabotage by undetectable secret sharing. Steganography has gained the limelight in the recent past and is formidable and belligerent as it involves embedding of secret data in either images, audio, video, etc., so unsuspectingly that even when intercepted cannot provide a hint to the hackers. However, to make it fool proof we cerebrate yet another algorithm in this paper which uses a combination of Hilbert and Moore Space Filling Curve (SFC) and the pixel indicator methodology, a steganographic tool to improve the randomness and cloak the scanning path such the adversary does not even spill on the clandestine information by an accident. Pixel Indicator (PI) technique uses the complexity of the color image which would be split into red, green and blue planes, respectively with one acting as an indicator, depending upon whose last two bits, data would be embedded in the other two planes. Thus instead of employing the usual raster scan, a random space filling curves (SFC) is used for embedding data into the pixels, because of which it would be impossible for eavesdropper to determine the path of embedding of data making it a highly robust system. The added advantages and enhancements provided by this technique can be observed by the readings of Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR) that have been obtained on implementing this algorithm.

Key words: Information hiding, space filling curve, pixel indicator method, steganography

INTRODUCTION
Ever since life took birth on this planet, there has been an inexplicable need for human beings to communicate, from the daily conversations to the exchange of clandestine information. And as the need for communication escalated and got satisfied by an equated technological advancement in the field of communication, there has also been an identical scenario in the area of information extraction, espionage and infringement. Thus it arouses the need for normal banking sector to the powerful government to put their surreptitious information behind a barrage of efficacious security systems. These beheld the methods for information hiding to evolve from sheer primitive procedures to top class complex algorithms classified as data hiding (Bender et al., 1996; Amirtharajan and Balaguru, 2009, 2010, 2011, 2012a, b; Amirtharajan et al., 2012; Cheddad et al., 2010).

Data hiding can be stratified into cryptography (Schneier, 2007; Salem et al., 2011), steganography and digital water marking (Stefan and Pabin, 2000) over a large basis.
Cryptography involves the coding of the source message into a different form which the sender and receiver can understand. But it lacks the main purpose when it is transmitted as illegal data seekers may be attracted by the coded form and thus the secrecy get lost. Steganography involves the coding of the source message into a different form called cover object, most probably an image, audio or video (Amirtharajan and Balaguru, 2009, 2010, 2011, 2012a, b; Zhu et al., 2011; Al-Frajat et al., 2010; Amirtharajan et al., 2012) such that it can be decoded only by the appropriate receiver with the correct key. Also it overcomes the deficit of cryptography in that; it doesn’t attract anyone on its transmission. This usually involves the replacement of the lower bit of each pixel of the cover image with the original message to form the stego image so as to reduce distortion (Thien and Lin, 2003; Chan and Cheng, 2004; Yang, 2008). Watermarking is mainly intended for copy right protection wherein the information to be sent is embedded in the digital signal in a way to verify its authenticity (Abdulsetah et al., 2009; Zhang et al., 2010; Abdulsetah et al., 2010; Zeki et al., 2011).

On going through the literature, it has been observed that steganography is one of the most effective methods for data hiding; with its strength being; it satisfies all three requirements of information hiding magic triangle of robustness, imperceptibility and high data payload:

- Robustness: Ability to withstand attacks and maintain integrity
- High payload: Maximum amount of data that can be hidden
- Imperceptibility: Ability to transmit data without attraction or suspicion of illegitimate party

Steganography is found to be the best as it satisfies all the above requirements fairly well than the other two. The usually employed technique in this method involves directly replacing the Least Significant Bit (LSB) of each pixel in the cover object thus causing very less distortion.

The steganographic methods may possibly be categorized into Spatial or Transform domain. In the former spatial domain of the cover objects are used to camouflage the secret data resulting in Stego object. Whereas the later employs the transformed domain of the cover objects are used to hide the clandestine information. The other classification is based on the types of cover object like Video, Audio, Image and Text (Shirali-Shahreza and Shirali-Shahreza, 2008; Al-Azawi and Fadhil, 2010). One more classification is based on the methods used on the chosen cover like Substitution (Amirtharajan and Balaguru, 2009, 2010, 2011, 2012a, b; Amirtharajan et al., 2012), Transform domain (Thanikaiselvan et al., 2011b) and Spread Spectrum (Kumar et al., 2011). Distortion, statistical and new cover generation (Xiang et al., 2011). The other side to break steganography called steganalysis are given by Xia et al. (2009) and Qin et al. (2009, 2010).

Image steganography is extensively used now-a-days through internet (Hmood et al., 2010a, b). It is the most common and well known method for high capacity, imperceptibility (Thanikaiselvan et al., 2011a, b). It could be classified as Least Significant Bit (LSB) substitution and pixel value differencing. In LSB substitution the least significant k-bits of target pixel in cover image are embedded with message bits but these methods will considerably introduce distortion in Stego image. To improve this, many new optimized LSB approaches have been suggested. Chan and Cheng (2004) proposed a simple LSB substitution method with Optimal Pixel Adjustment Process (OPAP) to reduce the Mean square error. Wang et al. (2001, 2008) methods offers high embedding capacity with good imperceptibility using adaptive Least Significant Bit (LSB) substitution along with pixel-value differencing (PVD).

Abbas Cheddad discusses a detailed survey on digital image steganography methods and its Classification (Cheddad et al., 2010). It also describes the differences among steganography;
watermarking and encryption and few other reviews on steganography is available (Amirtharajan et al., 2012; Rajagopalan et al., 2012; Janakiraman et al., 2012a,b; Thenmozhi et al., 2012). Pixel indicator based random image stego system proposed by Gutub et al. (2008) and Gutub (2010) and exploited by Amirtharajan (Amirtharajan et al., 2010, 2011; Padmaa et al., 2011). The number of bits embedding decided by most significant values (MSBs) (Amirtharajan et al., 2011), furthermore by calculating number of bits embedded through PVD (Amirtharajan et al., 2010) and so on. However, the embedding here is carried out using a simple Raster scan so may be assailable by third parties and information may be easily extracted.

To compensate this glitch, Hilbert or Moore SFC (Amirtharajan and Balaguru, 2009, 2010, 2012a; Zhao and Luo, 2012) based embedding can be used instead of Raster Scan (Thien and Lin, 2003; Chan and Cheng, 2004; Yang, 2008). The specific scan used can be kept confidential between sender and recipient and if hacked by an eavesdropper, would be nearly impossible to track out any information or find a pattern of data embedding. The entire cover image here would first have to be segregated into smaller blocks and then data should be embedded as per requirement. Also, an added advantage would be the inclusion of the Pixel Indicator (PI) technique (Amirtharajan et al., 2010, 2011; Padmaa et al., 2011), where one channel would be determined as an indicator and the specified amount of bits by user (say k bits) are then embedded in the other two channels depending upon the last two bits of the indicator channel.

In the spatial domain it is seen that LSB substitution is done mostly by Raster scan (Thien and Lin, 2003; Chan and Cheng, 2004; Yang, 2008) and sometimes by random scans (Amirtharajan and Balaguru, 2009, 2010, 2012b; Amirtharajan et al., 2012) to embed the information to help up the capacity, simplicity and time of implementation. In these random approaches all the pixels of the cover image have not been used for hiding secret data which in turn affect the payload besides good imperceptibility.

THE PROPOSED METHOD

Space filling curves (SFC) is a one dimensional curve which traverses through each and every point within a two dimensional space or image (Amirtharajan and Balaguru, 2009, 2010, 2012b; Zhao and Luo, 2012). SFC scans a pixel array which has a size of M×N pixels and while scanning, it will not retain the same direction but will turn around to embrace all the pixels at least and at most once. Hence the unpredictable traversing path of SFC through the image has been chosen to hide the secret message in the cover. In this scheme, both the sender and receiver can adapt a particular SFC so that there is no need to communicate the key and also providing a complicated traversing path for k-bit embedding which does not require any key.

Before considering the entire cover image for secret bit embedding, block of 4×4 pixels has been taken to implement Hilbert SFC and Moore SFC traversing path based stego technique by adapting a common traversing path for both the sender and receiver. After performing this process for a single 4×4 block, it has been extended to the full image by considering it as multiple of 4×4 blocks to cover up the entire $2^3$×$2^3$×3 pixels.

The traversing paths to embed secret data, based on Hilbert SFC and Moore SFC in gray scale are shown in Fig. 1 and 2 and Hilbert scan and Moore SFC in RGB in Fig. 3 and 4. Block diagram of the proposed embedding and extraction is shown in Fig. 5.

And the additional pixel indicator method used here involves selection of a channel as an indicator first and then depending upon the last two bits of the indicator, data is further embedded in the other two channels as given in Table 1.
Fig. 1: Hilbert scan

Fig. 2: Moore curve

Fig. 3: Hilbert scan in RGB
Fig. 4: Moore curve in RGB

Fig. 5: Block diagram of the proposed embedding and extraction

<table>
<thead>
<tr>
<th>Table 1: Meaning of indicator values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indicator</td>
</tr>
<tr>
<td>-----------</td>
</tr>
<tr>
<td>00</td>
</tr>
<tr>
<td>01</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>11</td>
</tr>
</tbody>
</table>

Thus if 'red' plane is selected as an indicator and its last two bits be '11' then k bits (as defined by user) of data are embedded in the blue and the green channel, respectively. Furthermore, the PI technique can be employed in three flexible methods:

- Red being the standard indicator for the entire cover image
- User-defined indicator
- Cyclic indicator for the cover
Case 1: k bit LSB with tri-colour random image steganography

Embedding algorithm:

**Inputs:** Cover image (C), secret data (D)

**Output:** Stego image (S)

1. Conversion of secret data (D) into binary
2. Split the cover image C into red (R), green (G) and blue (B) planes
3. For each pixel traversed by chosen SFC in R, do the following:
   3.1. Let b [0] = LSB of the current pixel in R
   3.2. Let b [1] = Next LSB of the current pixel in R
   3.3. If b = 00 then
       Go to next pixel.
   Else if b = 01 then
       Call k bit LSB Embedding to embed secret data in current pixel of G.
   Else if b =10 then
       Call k bit LSB Embedding to embed secret data in current pixel of B.
   Else
       Call k bit LSB embedding to embed secret data in current pixel of both G and B.
4. If all secret data is embedded, then
   Go to step-4
2. Store the resulting image as stego image (S) after applying OPAP

Recovery algorithm:

**Input:** Stego image (S)

**Output:** Secret data (D)

1. Split the stego image S into red (R), green (G) and blue (B) planes
2. For each pixel traversed by chosen SFC in R, do the following:
   2.1. Let b [0] = LSB of the current pixel in R
   2.2. Let b [1] = Next LSB of the current pixel in R
   2.3. If b = 00 then
       Go to next pixel.
   Else if b = 01 then
       Call k bit LSB recovery to recover secret data from current pixel of G.
   Else if b =10 then
       Call k bit LSB recovery to recover secret data from current pixel of B.
   Else
       Call k bit LSB recovery to recovery secret data from current pixel of both G and B.
3. Store the resulting recovered data as secret data (D)

Case 2: k bit LSB with custom-indicator-plane tri-color random image steganography

Embedding algorithm:

**Inputs:** Cover image (C), indicator-plane index (I) and secret data (D)

**Output:** Stego image (S)

1. Convert the secret data (D) into binary format
2. Split the cover image C into red, green and blue planes
3. If I = 1 then
Case 2: Continued


Else if \( l = 2 \), then


Else if \( l = 3 \), then


4. For each pixel traversed by chosen SFC in \( P[1] \), do the following:

4.1. Let \( b[0] \) = LSB of the current pixel in \( P[1] \)

4.2. Let \( b[1] \) = Next LSB of the current pixel in \( P[1] \)

4.3. If \( b = 00 \) then

Go to next pixel.

Else if \( b = 01 \) then

Call \( k \) bit LSB embedding to embed secret data in current pixel of \( P[2] \).

Else if \( b = 10 \) then

Call \( k \) bit LSB embedding to embed secret data in current pixel of \( P[3] \).

Else

Call \( k \) bit LSB embedding to embed secret data in current pixel of both \( P[2] \) and \( P[3] \).

4.4. If all secret data is embedded, then Go to step-5

3. Store the resulting image as Stego Image (S) after applying OPAP

**Recovery algorithm:**

**Input:** Stego image (S), indicator-plane index (I)

**Output:** Secret data (D)

1. Split the stege image S into red, green and blue planes. (R, G and B, respectively)

2. If \( I = 1 \) then:


Else if \( I = 2 \), then:


Else if \( I = 3 \), then:


3. For each pixel traversed by chosen SFC in \( P[1] \), do the following:

3.1. Let \( b[0] \) = LSB of the current pixel in \( P[1] \)

3.2. Let \( b[1] \) = Next LSB of the current pixel in \( P[1] \)

3.3. If \( b = 00 \) then

Go to next pixel.

Else if \( b = 01 \) then

Call \( k \) bit LSB recovery to recover secret data from current pixel of \( P[2] \).

Else if \( b = 10 \) then

Call \( k \) bit LSB recovery to recover secret data from current pixel of \( P[3] \).

Else

Call \( k \) bit LSB recovery to recover secret data in current pixel of both \( P[2] \) and \( P[3] \).

4. Store the resulting data as secret data (D).

Case 3: \( k \) bit LSB with cyclic-indicator-plane tri-colour random image steganography

**Embedding algorithm:**

**Inputs:** Cover image (C), secret data (D)

**Output:** Stego image (S) with secret data embedded in it.

1. Conversion of secret data (D) into binary representation.

2. Split the cover image C into red, green and blue planes.

3. Let index \( I = 1 \).
Case 3: Continued

4. For each pixel in P[1], do the following:
   4.1. If (i mod 3) = 1 then,
       \[ l[i] = 1 \]
       Else if (i mod 3) = 2 then,
       \[ l[i] = 2 \]
       Else
       \[ l[i] = 3 \]
   4.2. Set \( i = i+1 \)
5. Let index \( j = 0 \)
6. For each pixel traversed by chosen SFC in P[1], do the following:
   6.1. If \( l[j] = 1 \) then,
       Else if \( l[j] = 2 \), then
       Else if \( l[j] = 3 \), then
   6.2. Let \( b[0] = \) LSB of \( P[1] \)
   6.3. Let \( b[1] = \) Next LSB of \( P[1] \)
   6.4. If \( b = 00 \) then
       Go to next pixel
       Else if \( b = 01 \) then
       Call \( k \) bit LSB Embedding to embed secret data in \( P[2] \).
       Else if \( b = 10 \) then
       Call \( k \) bit LSB Embedding to embed secret data in \( P[3] \).
       Else
       Call \( k \) bit LSB Embedding to embed secret data in both \( P[2] \) and \( P[3] \).
   6.5. If all secret data is embedded, then
       Go to step-7
   Else
   \[ j = j+1 \]
7. Store the resulting image as Stego Image (S) after applying OPAP.

Recovery algorithm:

Input: Stego image (S)  
Output: Secret data (D)

1. Split the stego image S into red, green and blue planes. (R, G and B, respectively)
2. Let index \( i = 1 \)
3. For each pixel in P[1], do the following:
   4.1. If (i mod 3) = 1 then,
       \[ l[i] = 1 \]
       Else if (i mod 3) = 2 then,
       \[ l[i] = 2 \]
       Else
       \[ l[i] = 3 \]
   4.2. Set \( i = i+1 \)
4. Let index \( j = 0 \)
Case 3: Continued
5. For each pixel traversed by chosen SFC in $P[1]$, do the following:

5.1. If $l[2] = 1$ then,
   Else if $l[2] = 2$, then
   Else if $l[2] = 3$, then

5.2. Let $b[0] = \text{LSB of } P[1]$
5.3. Let $b[1] = \text{Next LSB of } P[1]$
5.4. If $b = 00$ then
   Go to next pixel
   Else if $b = 01$ then
   Call k bit LSB Recovery to embed secret data from $P[2]$. 
   Else if $b = 10$ then
   Call k bit LSB Recovery to embed secret data from $P[3]$. 
   Else
6. Store the resulting data as Secret Data (D)

RESULTS AND DISCUSSION

To evaluate the performance of our proposed method several experiments are performed. Figure 6 shows embedding and extraction flowchart. Four colour images are taken with size $256 \times 256$ as cover images which are shown in Fig. 7. Initially for varying $k = 1, 2, 3$ and 4 bit embedding performed and the values for $K = 4$ bit embedding is given in Table 2, 3 and 4 for method 1, 2 and 3 and the corresponding Stego covers for method 4 are given in Fig. 8.

To evaluate the performance of the proposed system MSE and PSNR have been computed for all the three methods. Peak Signal to Noise Ratio (PSNR) and Mean Square Error (MSE).

The PSNR is calculated using the equation:

$$\text{PSNR} = 10 \log_{10} \left( \frac{L}{\text{MSE}} \right) \text{dB}$$

(1)

where, $I_{max}$ is the intensity value of each pixel which is equal to 255 for 8 bit gray scale images.

The MSE is calculated by using the Eq. 2 given below:

$$\text{MSE} = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} (X_{ij} - Y_{ij})^2$$

(2)

where, $M$ and $N$ denote the total number of pixels in the horizontal and the vertical dimensions of the image $X_{ij}$ represents the pixels in the original image and $Y_{ij}$ represents the pixels of the stego-image.

From Table 2 it’s observed that Mahatma Gandhi cover has the maximum embedding capacity and minimum PSNR values. There is no change in the red plane histogram Fig. 9.
Fig. 6: Embedding and extraction flowchart

Fig. 7(a-d): Cover images; (a) Lena, (b) Baboon, (c) Gandhi and (d) Temple

Fig. 8(a-d): Stego output $k = 4$ bit for maximum embedding capacity; (a) Lena, (b) Baboon, (c) Gandhi and (d) Temple
Table 2: MSE PSNR values for method 1 k = 4 bit for maximum embedding capacity

<table>
<thead>
<tr>
<th>Cover images</th>
<th>Lena</th>
<th>Baboon</th>
<th>Gandhi</th>
<th>Temple</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSE red plane</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>MSE green plane</td>
<td>3.5752</td>
<td>3.5317</td>
<td>3.6997</td>
<td>3.5102</td>
</tr>
<tr>
<td>MSE blue plane</td>
<td>3.5816</td>
<td>3.5539</td>
<td>3.6543</td>
<td>3.548</td>
</tr>
<tr>
<td>PSNR red plane</td>
<td>8.000</td>
<td>8.000</td>
<td>8.000</td>
<td>8.000</td>
</tr>
<tr>
<td>PSNR green plane</td>
<td>42.5978</td>
<td>42.561</td>
<td>42.4845</td>
<td>42.6775</td>
</tr>
<tr>
<td>PSNR blue plane</td>
<td>42.8286</td>
<td>42.8286</td>
<td>42.5027</td>
<td>42.6909</td>
</tr>
<tr>
<td>No. of bits in green plane</td>
<td>132396</td>
<td>131576</td>
<td>132836</td>
<td>129768</td>
</tr>
<tr>
<td>No. of bits in blue plane</td>
<td>131332</td>
<td>131364</td>
<td>134092</td>
<td>131440</td>
</tr>
<tr>
<td>Total No. of bits embed</td>
<td>263628</td>
<td>262940</td>
<td>296888</td>
<td>291208</td>
</tr>
</tbody>
</table>

Table 3: Method 2 MSE PSNR values for k = 4 bit embedding and green plane as indicator (User choice)

<table>
<thead>
<tr>
<th>Cover images</th>
<th>Lena</th>
<th>Baboon</th>
<th>Gandhi</th>
<th>Temple</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSE red plane</td>
<td>3.1329</td>
<td>3.5176</td>
<td>3.6942</td>
<td>3.9831</td>
</tr>
<tr>
<td>MSE green plane</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>MSE blue plane</td>
<td>3.9109</td>
<td>3.8921</td>
<td>3.9653</td>
<td>3.8215</td>
</tr>
<tr>
<td>PSNR red plane</td>
<td>42.1419</td>
<td>42.9298</td>
<td>42.5484</td>
<td>42.8971</td>
</tr>
<tr>
<td>PSNR green plane</td>
<td>8.000</td>
<td>8.000</td>
<td>8.000</td>
<td>8.000</td>
</tr>
<tr>
<td>PSNR blue plane</td>
<td>42.9873</td>
<td>42.8151</td>
<td>42.0754</td>
<td>42.9271</td>
</tr>
<tr>
<td>No. of bits in green plane</td>
<td>131044</td>
<td>130828</td>
<td>131520</td>
<td>130664</td>
</tr>
<tr>
<td>No. of bits in blue plane</td>
<td>131684</td>
<td>130984</td>
<td>129136</td>
<td>133092</td>
</tr>
<tr>
<td>Total No. of bits embed</td>
<td>262278</td>
<td>261612</td>
<td>260656</td>
<td>263766</td>
</tr>
</tbody>
</table>

Fig. 9: Method 1 RGB histogram for temple, no change in red plane

From Table 3 it’s observed that Thanjavur Big Temple cover has the maximum embedding capacity and minimum PSNR values. There is no change in the green plane histogram (Fig. 10).

From Table 5 it’s observed that Baboon cover has the maximum embedding capacity and minimum PSNR values and the error are evenly distributed in all the planes (Fig. 11).
Table 4: Method 3 MSE PSNR values for k = 4 bit embedding and cyclic indicator

<table>
<thead>
<tr>
<th>Cover images</th>
<th>Lena</th>
<th>Baboon</th>
<th>Gandhi</th>
<th>Temple</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSE red plane</td>
<td>2.4387</td>
<td>2.3702</td>
<td>2.5728</td>
<td>2.3143</td>
</tr>
<tr>
<td>MSE green plane</td>
<td>2.3066</td>
<td>2.3255</td>
<td>2.4212</td>
<td>2.3005</td>
</tr>
<tr>
<td>MSE blue plane</td>
<td>2.3389</td>
<td>2.3619</td>
<td>2.3599</td>
<td>2.3764</td>
</tr>
<tr>
<td>PSNR red plane</td>
<td>44.259</td>
<td>44.2382</td>
<td>44.0267</td>
<td>44.4866</td>
</tr>
<tr>
<td>PSNR green plane</td>
<td>44.501</td>
<td>44.4657</td>
<td>44.2904</td>
<td>44.4957</td>
</tr>
<tr>
<td>PSNR blue plane</td>
<td>44.441</td>
<td>44.3981</td>
<td>44.4926</td>
<td>44.3716</td>
</tr>
<tr>
<td>No. of bits in red plane</td>
<td>85640</td>
<td>85652</td>
<td>85112</td>
<td>85768</td>
</tr>
<tr>
<td>No. of bits in green plane</td>
<td>85696</td>
<td>86504</td>
<td>86552</td>
<td>84956</td>
</tr>
<tr>
<td>No. of bits in blue plane</td>
<td>85730</td>
<td>86552</td>
<td>85132</td>
<td>86436</td>
</tr>
<tr>
<td>Total No. of bits embed</td>
<td>256776</td>
<td>257108</td>
<td>256706</td>
<td>257160</td>
</tr>
</tbody>
</table>

Table 5: Comparative estimation parameters of the proposed embedding scheme-III

<table>
<thead>
<tr>
<th>Cover image</th>
<th>Channel I red</th>
<th>Channel II green</th>
<th>Channel III blue</th>
<th>BFF (bits per pixel)</th>
<th>Maximum embedding capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>Proposed</td>
<td>2.4387</td>
<td>44.259</td>
<td>2.3066</td>
<td>44.501</td>
</tr>
<tr>
<td>Padma[10]</td>
<td>1.227</td>
<td>47.24</td>
<td>1.3641</td>
<td>46.782</td>
<td>1.02</td>
</tr>
<tr>
<td>Amir[11]</td>
<td>1.68</td>
<td>45.89</td>
<td>1.57</td>
<td>46.18</td>
<td>1.52</td>
</tr>
<tr>
<td>Amir[12]</td>
<td>1.2906</td>
<td>47.0230</td>
<td>1.2374</td>
<td>47.2099</td>
<td>1.2049</td>
</tr>
<tr>
<td>Baboon</td>
<td>Proposed</td>
<td>2.3702</td>
<td>44.3829</td>
<td>2.3325</td>
<td>44.4657</td>
</tr>
<tr>
<td>Amir[11]</td>
<td>2.61</td>
<td>43.95</td>
<td>2.65</td>
<td>43.88</td>
<td>2.72</td>
</tr>
<tr>
<td>Amir[12]</td>
<td>1.5540</td>
<td>46.2162</td>
<td>1.5544</td>
<td>46.2151</td>
<td>1.5904</td>
</tr>
<tr>
<td>Gandhi</td>
<td>Proposed</td>
<td>2.5728</td>
<td>44.0207</td>
<td>0.6738</td>
<td>44.2204</td>
</tr>
<tr>
<td>Padma[10]</td>
<td>1.348</td>
<td>46.83</td>
<td>2.4212</td>
<td>47.025</td>
<td>1.2478</td>
</tr>
<tr>
<td>Amir[11]</td>
<td>1.34</td>
<td>46.83</td>
<td>1.30</td>
<td>47.07</td>
<td>1.24</td>
</tr>
<tr>
<td>Amir[12]</td>
<td>3.2721</td>
<td>42.9832</td>
<td>3.2944</td>
<td>42.9530</td>
<td>3.1356</td>
</tr>
<tr>
<td>Temple</td>
<td>Proposed</td>
<td>2.3143</td>
<td>44.4896</td>
<td>2.3096</td>
<td>44.4957</td>
</tr>
<tr>
<td>Padma[10]</td>
<td>1.853</td>
<td>45.45</td>
<td>1.76</td>
<td>45.66</td>
<td>1.63</td>
</tr>
<tr>
<td>Amir[11]</td>
<td>1.85</td>
<td>45.45</td>
<td>1.76</td>
<td>45.66</td>
<td>1.63</td>
</tr>
<tr>
<td>Amir[12]</td>
<td>1.1159</td>
<td>47.6547</td>
<td>1.1062</td>
<td>47.6532</td>
<td>1.1240</td>
</tr>
</tbody>
</table>

COMPARATIVE RESULTS

Complexity analysis: The complexity of the proposed system is also good. This method divides the cover image into 4×4 blocks. So there will be 4096 such blocks, so by taking one red, green, blue 4×4 block and use Hilbert or Moore SFC for traversing and if they are selected randomly then there are 4096!*4096!*4096! ways to select one red, green, blue block. Pixel indicator additional increase the randomness assuming 25% on each case like 00, 01, 10 and 11 which decides the embedding capacity with secret data encrypted with AES. Then the complexity would be 4096!*3*5*2^128 Furthermore, first the indicator can be selected in three ways. If the LSBs of the first indicator is not zero then embedding is done.

- So the probability of embedding is 3/4
- The cover image is divided into 1024×4 blocks of 4×4 pixels
- Two different scan paths for random embedding process can be adopted Hilbert or Moore SFC 2 ways

135
Fig. 10: Method 2 RGB histogram for temple, no change in green plane

Fig. 11: Method 3 histogram for temple, errors are evenly distributed

- The PRN scheme has been adopted to select the block of $1024^4$, $NpR = N!(N-r)! = 1024^4p1024^4 = 4096!$
- The starting point of each $4 \times 4$ SFC has been changed. 16 different paths can be selected by having 16 different starting points in the selected SFC
- So total complexity for embedding/or hacking at least one bit is $2^{128^3/43^2/2^4096!^316$.
- Which is far better than Padma et al. (2011) $2^{64^3/7*0.5^2*2}$ and Amirtharajan et al. (2010) is only $2^{64^3/3^2}$ and Amirtharajan et al. (2011) is $2^{128^3/2^4/3*(8+8/3+8/3+8)$
CONCLUSION

Thus it is seen there are myriad of methodologies for incorporating data security and impregnability in our day-to-day life. While there may be a plethora of techniques to incur the basic objective of information hiding involving imperceptibility, robustness, high payload and security, it is seen that not all of them may obtain it simultaneously. Especially the robustness factor seems to be compromised with the routine techniques being employed, with the hackers too being continually educated with them. Thus this proposed methodology brings about a fantabulous shift from the mundane methods, by involving Hilbert or Moore space filling curves for traversing through the pixels with more integrity obtained by employing pixel indicator (PI) concurrently. The added advantage in the proposed method is the ability to determine the indicator; either by the user or a cyclic one or a fixed one. To re-enforce the entire objective of security, the data is scrambled using DCT before embedding. With 30 dB being fixed as the threshold PSNR value for human visual system, it is seen that this technique on embedding causes very little degradation and provides high imperceptibility 42 dB and above for all the cases. Thus it is observed that this algorithm involving random curves such as Hilbert or Moore SFC for traversing along with PI, is a boon for transfer of information, because when it is intercepted by hackers too, is an absolute nightmare to decode as they wouldn’t have a clue as where as to start from. Overall, this method caters to all the requirements of information hiding in a single package.

REFERENCES


(let) decide choosy pixel embedding for stego. Proceedings of the International Conference on
Thanikaiselvan, V., S. Kumar, N. Neelima and R. Amirtharajan, 2011b. Data battle on the digital
Thenmozhi, K., P. Praveenkumar, R. Amirtharajan, V. Prithviraj, R. Varadarajan and
Zeki, A.M., A.A. Manaf, A.A. Ibrahim and M. Zamani, 2011. A robust watermark embedding in
Zhang, Y., Z.M. Lu and D.N. Zhao, 2010. A blind image watermarking scheme using fast