Data Hiding approach based on Eight-Queens Problem and Pixel Mapping Method

Abhishek Bansal¹, Sunil Kumar Muttoo² and Vinay Kumar³

¹Assistant Professor, Indira Gandhi National Tribal University, Amarkantak M.P.  
E-mail: bansalabhishek28@gmail.com

²Associate Professor, Department of Computer Science, University of Delhi  
E-mail: skmuttoo@cs.du.ac.in

³Professor, Vivekananda Institute of professional Studies, GGSIPU, Delhi  
E-mail: vinay.kumar@nic.in, vinay5861@gmail.com

Abstract

A new steganographic algorithm is presented using pixel mapping method based on eight queens and number of ones in pixel intensity value. In our method, we are sequentially selecting 8x1 pixel blocks to embed message bit. The method works with randomizing the bit selection in the 8x1 pixel block using eight queens’ solutions. This approach finds relationship between secret message and cover image according to the pixel mapping table. In this paper, we have presented a secure image steganography technique with acceptable levels of imperceptibility and distortion in the cover image. In presenting algorithm, security is considered by the randomize selection of eight queen’s solutions based on a seed value. The presented algorithm not only provides high level security, but also produces high capacity and good imperceptibility. The algorithm has been tested with different image file formats like BMP, TIFF and PNG.

Keywords: Steganography, Pixel Mapping Method, information hiding, Random Pixel Selection, LSB

1. Introduction

Secrecy of communication is essentially required to maintain the sanctity of privacy. Steganography is one of tested technique of secret communication. The word steganography is a combination of two Greek words – STEGANOS mean Covered and GRAPHIE mean Writing. Steganography is an art and science of secret communication which is used to conceal secret information within a cover media like text, image, audio and video files in such a way that prevents an unauthorized user to detect hidden message [9].

Steganography methods could be classified into two categories. The first categories employ the domain of a host image to camouflage secrete message i.e., The secret message is directly embedded into the pixels of the host image and second employs the transformed domain, such as DCT, DWT of a host image to hide secret data [15]. The process of embedding is briefly explained as below.

Let C be a cover carrier image I and S be the stego image. Let K represent an optional key (a seed used to generate a pseudorandom noise) and let M be the message we want to hide into cover. If Eₘ is embedding process (function) and Eₓ be extraction process [1] then

\[ E_m(C, K, M) = S \]
\[ E_x(E_m(C, K, M)) = M \]
There are different methods for data hiding such as least significant bit methods, pixel-value difference methods, histogram modification and pixel mapping methods that use spatial domain. Many works have been done in this area and many methods have been developed and reported [5, 6, 8, 10, 13, 21, 22].

In 2010 Bhattacharyya et al. proposed an image transformation technique in [2, 16-18] based on Pixel Mapping Method (PMM). Pixel mapping method is an embedding technique in pixel intensity using some mathematical function based on seed value and its 8 neighbours are selected in counter clockwise direction. This approach provides better embedding capacity and PSNR value over other spatial methods. In 2011 Bhattacharyya S., et al., proposed another approach [15] based on pixel mapping method in 8x8 blocks of the each selected plane. The integrated approach of PMM and Bit Plane Complexity Segmentation (BPCS) produces a robust image based steganography method which is independent of the nature of the data to be hidden and produces a stego image with minimum degradation. In 2004 Potdar, et al., [14] proposed a technique to hide data by changing the gray level values of the gray scale image pixels called gray level modification method.

In this paper, we have presented an efficient information hiding technique based on pixel mapping relationship using eight-queen problem [4, 7] and number of ones in intensity value. A schematic diagram of the proposed method is shown in Figure 1. Our presented pixel mapping method is applicable on gray level modification. This work is an extension to our previous work [3, 12] where data hiding techniques are based on eight queen’s solutions. The results of the presented method have been compared with our previous technique. The results show better stego image quality and high embedding capacity.

The paper is organized in seven sections. Section 2 presents the basics of eight-queen problem. Section 3 describes pixel mapping method based on eight queen’s problems. Embedding and Extracting Algorithm is discussed in Section 4. Section 5 defines various experimental results. Comparison with our previous methods is discussed in Section 6. Steganalysis of the presented approach is discussed in Section 7 together with the concluding part of the paper.

2. Eight Queen’s Method
In chess, a queen can move horizontally, vertically, or diagonally. A chess board has 8 rows and 8 columns. Two questions are related to this problem. One is “what is the maximum number of queens that can be placed on \( n \times n \) chessboard such that no two queens attack one another?” And the other is “how many ways can one put \( n \) queens on \( n \times n \) chessboard so that no two queen attacks each other?” [7].

The answer is proposed by Madachy [11, 16]. The number of different ways the 8 queens can be placed on an 8*8 chessboard so that no two queens may attack each other is 92. The 12 distinct solutions as shown in Figure 2 and the remaining 80 are generated by rotation and reflection [11, 16].

![Figure 2. 12 Distinct Solution of 8×8 Queen’s Problem](image)

In this paper, we used eight queen solutions for extracting bits from 8×8 binary bit planes as shown in Figure 3. First bit will be in 1st row and 3 columns. Second bit will be in the second row and 6th columns. Third bit will be in the 3rd row and 2nd column and so on.

![Figure 3. Mapping with 8 Queen Solution’s in 8×8 Bit Plane](image)

In 8×8 chessboard, we can place 8 queen’s 92 different ways in non attacking position. We can represent board position 8×8 vector as \([3, 6, 2, 7, 1, 4, 8, 5]\) and store
in array Q. Similarly calculate all permutations of the queen’s solution and store in array Q. The size of array Q is 92×8.

3. Pixel Mapping Method

In the spatial domain, Information hiding based on mathematical relationships of pixel intensity is called pixel mapping methods. In our presented algorithm, pixel mapping is based on relationship of eight-queen solution and number of ones in intensity of pixel as shown in the Table 1. In presented method, data embedding is done by mapping two bits of secret message in each pixel based on following features.

(i) Consider an 8 x 1 pixel block from cover image and then masked with a solution of the eight queen’s solutions based on seed to find 8-bit string corresponding to each pixel in a block. If the secret message bit corresponding to pixel is matched with a queen’s solution bit, then no need to change in least significant bit and if no matching then least significant bit will be 1 as shown in Figure 4.

(ii) Other 8 bits from secret message is mapped according to the number of ones in pixel intensity of the image. If number of ones in pixel even then, consider message bit is zero otherwise one as shown in Figure 5.

Table 1. Mapping Technique for Embedding of Two Bit in each Pixel of Cover Image

<table>
<thead>
<tr>
<th>Pair of message bit</th>
<th>LSB bit</th>
<th>Queen’s solution bit</th>
<th>No. of one’s in pixel</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
<td>Queen bit 0</td>
<td>Even</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Complement of Queen bit 1</td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>0</td>
<td>Queen bit 0</td>
<td>Odd</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Complement of Queen bit 1</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>Queen bit 1</td>
<td>Even</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Complement of Queen bit 0</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>Queen bit 1</td>
<td>Odd</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Complement of Queen bit 0</td>
<td></td>
</tr>
</tbody>
</table>

Let Secret message bits are 10101001

8×1 pixel block

<table>
<thead>
<tr>
<th>Pixel 1</th>
<th>Pixel 2</th>
<th>Pixel 3</th>
<th>Pixel 4</th>
<th>Pixel 5</th>
<th>Pixel 6</th>
<th>Pixel 7</th>
<th>Pixel 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 0 1 0 1 0 0 1 0</td>
<td>0 0 1 0 1 1 1 1 0</td>
<td>1 1 1 1 0 1 1 1 1</td>
<td>0 0 1 0 0 0 0 0 0</td>
<td>0 1 0 1 1 0 1 1 0</td>
<td>0 0 0 0 1 0 0 1 0</td>
<td>1 0 1 1 0 0 0 1 1</td>
<td>0 1 0 0 0 0 1 1 0</td>
</tr>
<tr>
<td>8×8 bit plane with LSB 0 before mapping</td>
<td>8 queen’s solutions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4. Data Hiding Approach

Our presented data hiding approach is based on pixel mapping using eight queen’s solution and number of one’s in pixel intensity. To implement algorithm we need gray level modification for mapping with a queen’s solution. We take an image as the cover. Cover image contains a fixed number of rows and columns of pixels. Pixels are the smallest individual element in an image, holding quantized values that represent the brightness of a given colour at any specific point. For mapping we first convert all quantized values as even by modification of least significant bit either 0 or 1 as shown below.

Image Pixels = \{23, 14, 79, 155, 134, 125, 122\}
Modified Pixels = \{22, 14, 78, 154, 134, 124, 122\}

After gray level modification cover image is divided into block of 8 x 1 pixels and then masked with solutions of the eight queen’s solution to make an 8-bit string corresponding to each pixel in block as describe in embedding algorithm.

4.1. Embedding Algorithm

In presented method we take an image as cover and hide a secret message within the image based on pixel matching method. The cover image is a sequence of 8x1 pixel intensity, i.e., 8x8 bit binary planes. Our approach of pixel mapping method is based on
eight queens’ problem and number of ones in pixel intensity. An 8-queen problem has 92 solutions as explained in the section 2. First pixel mapping in 8 x 8 binary bit planes will be based on any one 8 queen solution based on seed value and another is based on number of ones in pixel intensity if it is even then message bit 0 otherwise it will be 1.

Let the message be M. M is a string of characters. While embedding messages M in an image cover C, C is divided into 8x1 bytes i.e. consider 8x8 bit plane blocks. For security, we take a large integer as a stego-key k1. For a randomized selection of the queen’s solution in 8x8 bit plane, we extract MSB bits and store it as decimal d. We compute k1 modulo d. Let it be R1. Again, we compute R1 modulo 92. Let us store the result in R2. 8 x 8 bit plane block is masked one by one with 8 solutions of eight queen’s problems from R2 and perform pixel matching as explained in section 4. A similar procedure will be applied to the next block. The stepwise algorithm is outlined below.

**Embedding Algorithm**

**Step 1:** Take cover image C and convert all pixel as even number (all LSB bit 0).

**Step 2:** Cover image C divide it into 8x1 byte blocks which is equivalent to 8x8 bit planes ci where 1 ≤ i ≤ n. Initialize i with 1.

**Step 3:** Initialize an array Q with all the 92 solutions of 8-queens problem.

**Step 4:** Take the secret message M and store each of its m characters Bj as 8-bit ASCII number.

**Step 5:** Take a large integer as a Key k1.

**Step 6:** Take first 8x8 bits block ci of cover C and do following -

6.1 Extract MSB bit from 8x8 bits block ci and convert it into decimal number d1

6.2 Find mod (k1, d1) and store result in R1.

6.3 Find mod (R1, 92) and store result in R2.

6.4 Extract 8 bits from ci corresponding to Q[R2], let it be E1

6.5 Take first character B0 do the following:

6.5.1 Determine R3= E1 XOR B0

6.5.2 Result R3 is embedded into LSB position of block ci

6.5.3 Take second character B_{j+1}, convert it into 8 bit binary sequence and do the following for each pixel in 8*1 byte block ci -

6.5.3.1 Take a first pixel p_i from ci and count number of ones in pixel intensity

6.5.3.2 If number of one’s in p_i is even and message character B_{j+1} first bit is 0 do nothing

Else If 2nd LSB bit in p_i is the part of queen’s solution Then change 3rd LSB bit in p_i such a way number of one’s becomes even

Else change 2nd LSB bit in p_i such a way number of one’s become even
6.5.3.3 If number of one’s is odd and message character $B_{j+1}$ first bit is 1 do nothing

Else if 2nd LSB bit in $p_i$ is the part of queen’s solution Then change 3rd LSB bit in $p_i$ such a way number of one’s become odd

Else change 2nd LSB bit in $p_i$ such a way number of one’s become odd

6.5.3.4 Go to step 6.5.3.1 for embedding other bits based on number of ones in others pixel of 8 x 1 blocks

**Step 7:** Go to step 5 and look for hiding next characters $B_{j+2}$ and $B_{j+3}$ in the next 8x8 bits block $c_{i+1}$.

### 4.2. Extraction Algorithm

Once the stego image is received at the receiver end, the extraction of hidden message is initiated by using the similar algorithm as outline below.

**Step 1:** Divide Stego image $C$ into 8×1 pixel block i.e., 8×8 bit binary block, say $c_i$. Let the number of block in $C$ be $n$.

**Step 2:** Initialize an array $Q$ with all the 92 solutions of 8-queens problem. Initialize message index $j = 1$ and message array $M$.

**Step 3:** Get Key $k_1$.

**Step 4:** Take first 8x8 bits block $c_i$ of cover $C$ and do following

4.1 Extract MSB bit from 8x8 bits block $c_i$ and convert it into decimal number $d_i$.

4.2 Find mod ($k_1$, $d_i$) and store result in $R_1$.

4.3 Find mod ($R_1$, 92) and store result in $R_2$.

4.4 Extract 8 bits from $c_i$ corresponding to $Q[R2]$ if LSB position bit 1 then consider 0, let it be $E_1$.

**Step 5:** Message Extraction using Pixel Mapping in block $c_i$ of cover $C$ using 8 queen’s solution.

5.1 Extract 8 bits in LSB position in block $c_i$, let it be $E_2$

5.2 Extract LSB bit in $E_1$, let it be $E_3$

5.3 Determine message $M1 = E_1 \text{ XOR } E_2$

**Step 6:** Convert results $M1$ into its equivalent ASCII code and store $M [j]$ ; increment $j$.

**Step 7:** Message Extraction using Pixel Mapping in block $c_i$ of cover $C$ based on number of one’s in pixel intensity.

7.1 For each pixel in block $c_i$ of cover $C$ do the following -

7.1.1 Find $\text{cnt} = \text{count number of one’s in pixel of block } c_i \text{ of cover } C$

7.1.2 If ($\text{cnt} \% 2 == 0$) then $m = 0$ Else $m = 1$

7.1.3 $M2 = M2 \| m$ (Concatenation)

**Step 8:** Convert results $M2$ into its equivalent ASCII code and store $M [j]$ and increment $j$. 
5. Experimental Result

The performance of the presented method has been evaluated and compared with other spatial domain data hiding techniques [3, 12] technique based on eight queen’s solutions. More than 300 standard gray scale images of size 256 *256 have been used to test the presented algorithm to hide secret message. The results are tabulated for eight standard images used as a cover and it is shown in Figure 7. The quantitative measure PSNR is used for comparison between the original cover image and stego image of dimension m x n. Peak Signal Noise Ratio (PSNR) is defined as below:

\[
PSNR = 10 \times \log_{10} \left( \frac{255^2}{MSE} \right) \text{ dB},
\]

\[
MSE = \frac{MSE_r + MSE_g + MSE_b}{3},
\]

\[
MSE_r = \frac{M - 1}{N - 1} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left( X_r(i, j) - I_r'(i, j) \right)^2,
\]

where \(X_r(i, j)\) and \(I_r'(i, j)\) represent the pixel values on the original image and the stego image located at \((i, j)\) respectively. M and N represent the height and width respectively, of the images. The PSNR value of greater than 30dB is considered as safe value for retaining the similarity between cover and stego images. The presented method is based on pixel mapping method. In Table 2 lists the PSNR and Embedding capacity based on mapping with eight queens method and Table 3 lists the PSNR and Embedding capacity based on mapping with eight queens method and number of one’s in pixel intensity.

![Figure 7. Cover Image](image)

(a) Lena, (b) Baboon, (c) Jet, (d) Peppers, (e) Mountain, (f) Pentagon, (g) Cameramen, (h) Fishing boat
Table 2. PSNR and Embedding Capacity of Various Tested Image based on Eight Queen Mapping

<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>Image</th>
<th>PSNR</th>
<th>Embedding Capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Lena.bmp (24 bit)</td>
<td>51.2687</td>
<td>196608 bits</td>
</tr>
<tr>
<td>2</td>
<td>Baboon.bmp (24 bit)</td>
<td>51.2870</td>
<td>196608 bits</td>
</tr>
<tr>
<td>3</td>
<td>Pepper.bmp (24 bit)</td>
<td>51.3284</td>
<td>196608 bits</td>
</tr>
<tr>
<td>4</td>
<td>Mountain.png (24 bit)</td>
<td>51.2451</td>
<td>196608 bits</td>
</tr>
<tr>
<td>5</td>
<td>Jet.bmp (24 bit)</td>
<td>51.2698</td>
<td>196608 bits</td>
</tr>
<tr>
<td>6</td>
<td>Pentagon(8 bit)</td>
<td>51.2603</td>
<td>65536 bits</td>
</tr>
<tr>
<td>7</td>
<td>Camramen.tif (8 bit)</td>
<td>51.2658</td>
<td>65536 bits</td>
</tr>
<tr>
<td>8</td>
<td>Fishingboat.bmp (8 bit)</td>
<td>51.2743</td>
<td>65536 bits</td>
</tr>
</tbody>
</table>

Table 3. PSNR and Embedding Capacity of Various Tested Image based on Eight Queens and Number of One’s in Pixels

<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>Image</th>
<th>PSNR</th>
<th>Embedding Capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Lena.bmp (24 bit)</td>
<td>44.4510</td>
<td>393224 bits</td>
</tr>
<tr>
<td>2</td>
<td>Baboon.bmp (24 bit)</td>
<td>44.4819</td>
<td>393224 bits</td>
</tr>
<tr>
<td>3</td>
<td>Pepper.bmp (24 bit)</td>
<td>44.4204</td>
<td>393224 bits</td>
</tr>
<tr>
<td>4</td>
<td>Mountain.png (24 bit)</td>
<td>44.3419</td>
<td>393224 bits</td>
</tr>
<tr>
<td>5</td>
<td>Jet.bmp (24 bit)</td>
<td>44.7144</td>
<td>393224 bits</td>
</tr>
<tr>
<td>6</td>
<td>Pentagon(8 bit)</td>
<td>44.3497</td>
<td>131080 bits</td>
</tr>
<tr>
<td>7</td>
<td>Camramen.tif (8 bit)</td>
<td>44.5412</td>
<td>131080 bits</td>
</tr>
<tr>
<td>8</td>
<td>Fishingboat.bmp (8 bit)</td>
<td>44.6275</td>
<td>131080 bits</td>
</tr>
</tbody>
</table>

6. Comparison with other Recent Schemes

Figure 8 and Figure 9 depict the comparison of embedding capacity in bits and image quality in PSNR of the presented method with that of the existing data hiding method based on eight queens [3, 12] for different image. Figure 8 shows that the embedding capacity of our presented method is higher than our previous method based on eight queens. Figure 9 shows that the quality of image is respect to PSNR is acceptable in our presented method. Thus we have improved our previous work and derived better performance. It further confirms that the presented scheme achieves high capacity with acceptable distortion.
7. Conclusion

Steganalysis is a technique of discovering any covert message that is embedded in the stego medium. In cryptanalysis, portions of the plaintext and portions of the cipher text are analyzed. In Steganalysis, comparisons are made between the cover-object, the stego-object and possible portions of the message. The algorithm presented is robust against known cover attack where both stego object and original cover are available to steganalyst because message is not directly embedded in cover but mapped with eight queen’s solution. The selection of eight queens is based on key. Similarly A known message attack is also difficult in presented method because the complexity involved in finding eight queens solutions and mapping with mathematical relationship.

In this paper, we have presented an efficient extension of data hiding approach based on eight queen’s solution by using pixel mapping method. The method provides high capacities at small distortion. Images stored in BMP, PNG and TIFF file formats is used to test the hiding algorithm. These file formats are taken for illustrative purpose only. Selection of eight queen solution based on seed value provides security in addition to the privacy provided by embedding algorithm. The seed value is based on key.
The approach can be further improved by using optimization algorithm such as particle swarm optimization (PSO), evolutionary algorithm and other neural network approach to enhance embedding capacity maintaining robustness and imperceptivity.

Acknowledgements

We are very grateful to all those who have been constantly encouraging and providing required resources for such scientific research work besides the regular work which we are doing in our respective departments. We are thankful to our colleagues and seniors for their continuous support and encouragements and to all those who provided suggestions for improvement of this work.

References

Authors

Abhishek Bansal, he did his MCA from Dr. Bhim Rao Ambedkar University, Agra, India in 2004. He is currently working as an Assistant Professor in the department of Computer Science, Indira Gandhi National Tribal University, M.P. India. He is pursuing PhD in Information Security from Delhi University, India. He has working experience more than 5 years. His research interests include digital Watermarking, data hiding techniques and image processing.

Sunil K. Muttoo, he is Associate Professor in Department of Computer Science, University of Delhi, India. He completed his M. Tech from IIT Kharagpur and Ph.D. from University of Delhi, India. His specialization is coding theory, information hiding.

Vinay Kumar, he is a Professor in Vivekananda Institute of Professional Studies, Delhi. Earlier he worked as Scientist in National Informatics Centre, MoCIT, GOI. He completed his Ph. D. in Computer Science from University of Delhi and MCA from JNU, Delhi. He has authored a book on Discrete Mathematics and contributed many research papers to refereed journals and conferences. His areas of interest are graph algorithm, steganography, data security, data mining and e-governance.