Steganography Using Sudoku Puzzle

Conference Paper · October 2009

DOI: 10.1109/ARTCom.2009.116 · Source: DBLP

Shanta Rangaswamy
Rashtreeya Vidyalaya College of Engineering

1

2009 International Conference on Advances in Recent Technologies in Communication and Computing

Steganography using Sudoku Puzzle

Roshan Shetty B R, Rohith J, Mukund V, Rohan Honwade
Final year B.E. (CSE)
R. V. College of Engineering
Bangalore, India
mukundv18@gmail.com

Shanta Rangaswamy
Asst. Professor, Department of Computer Science
R. V. College of Engineering
Bangalore, India
shanta_suthik@rediffmail.com

Abstract- **Steganography is the art of concealing the existence of information within seemingly harmless carriers. It is a method similar to covert channels, spread spectrum communication and invisible inks which adds another step in security. A message in cipher text may arouse suspicion while an invisible message will not. A digital image is a flexible medium used to carry a secret message because the slight modification of a cover image is hard to distinguish by human eyes. In this paper, we propose a revised version of information hiding scheme using Sudoku puzzle. The original work was proposed by Chang et al. in 2008, and their work was inspired by Zhang and Wang's method and Sudoku solutions. Chang et al. successfully used Sudoku solutions to guide cover pixels to modify pixel values so that secret messages can be embedded. Our proposed method is a modification of Chang et al's method. Here a 27 X 27 Reference matrix is used instead of 256 X 256 reference matrix as proposed in the previous method. The earlier version is for a grayscale image but the proposed method is for a colored image.** ***(Abstract)***

Keywords- **Steganography, Embedding, Extraction, data hiding, Reference matrix, RGB Image, pixels, secret data.** ***(Keywords)***

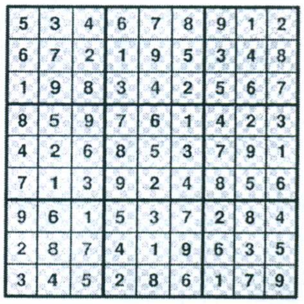
I. INTRODUCTION

Using digital images as cover media to conceal secret data is an important issue for secret data delivery applications. From the target of image modification, information hiding techniques can be classified into three domains, namely spatial domain [1], compressed domain [2], and transformed domain [3]. In spatial domain, more redundant spaces are available to secret data embedding so high embedding capacity can be achieved, and less time is needed for embedding and extracting procedures. However, information hiding schemes in spatial domain are vulnerable to common attacks such as statistical stegoanalysis. Two important factors needed to consider when we are designing a new information hiding scheme are embedding capacity (i.e. the number of secret bits can be embedded into one cover pixel) and visual quality of stego images (i.e. image distortion). Desirably, one would want to achieve high embedding capacity and good visual quality. However, embedding capacity and visual quality are inversely proportional to each other. That is, if embedding capacity is increased, then visual quality is decreased and vice versa. Thus, a tradeoff between embedding capacity and visual quality is made by users for different applications. Section 2 of this paper deals with the literature survey and related work, section 3 with the proposed method of data embedding and data extraction followed by conclusion and other further recommendations of the project.

The main aim of this paper is to improve the efficiency of the previously proposed method so as to make it possible to be implemented in the normal desktop computer with limited main memory and also to extend the digital media cover image from grey scale to RGB image (colour image).

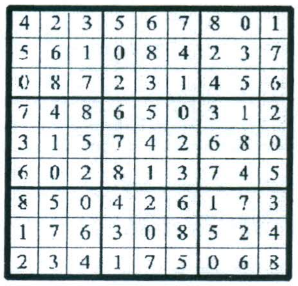
II. RELATED WORK

In this section, we will briefly describe Chang et al.'s [4] steganographic scheme based on Sudoku solutions. The central idea of Chang et al.'s method is to modify the selected pixel pairs in the cover image based on a specially designed reference matrix M to insert secret digits. For an 8-bit grayscale cover image, the size of the reference matrix M is designed to be 256× 256. To construct a reference matrix M, a "tile" matrix T is constructed first by subtracting every digit in a Sudoku puzzle by one, so that the digits in matrix T ranged from 0 to 8, as shown in Figures 1 and (b). The reference matrix M is then consisting of an $m \times m$ tiling of copies of T, where $m = \lfloor 256/9 \rfloor + 1$.



5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

Fig 1. Sudoku solution



4	2	3	5	6	7	8	0	1
5	6	1	0	8	4	2	3	7
0	8	7	2	3	1	4	5	6
7	4	8	6	5	0	3	1	2
3	1	5	7	4	2	6	8	0
6	0	2	8	1	3	7	4	5
8	5	0	4	2	6	1	7	3
1	7	6	3	0	8	5	2	4
2	3	4	1	7	5	0	6	8

Fig 2. Sudoku solution after subtracting 1

978-0-7695-3845-7/09 $26.00 © 2009 IEEE
DOI 10.1109/ARTCom.2009.116

IEEE computer society

-5-

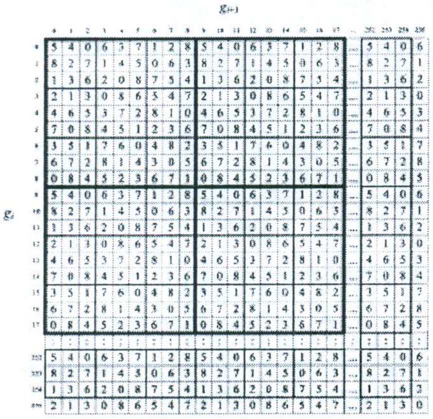
Note that matrix M has to be truncated to 256× 256. An example of a reference matrix M for a given solution of a Sudoku puzzle is shown in Fig 3.

A. *Data embedding*

In the data embedding phase, Chang et al. first convert the secret bit stream into secret digits in the base-9 numeral system, and then embed these secret digits into the cover image. Suppose the converted secret digits are denoted by $S=s_1s_2s_3....s_n$, where n is the total number of converted secret digits and $s_k \in [0,8]$, $1 \leq k \leq n$. In order to embed secrete digits; the cover image is partitioned into R non-overlapping blocks of size 1×2. This is done because a 8 bit grey scale image is used each pixel represents a grey scale which is 8 bits. Therefore two pixel values are required to represent the co-ordinate values. As it is known that each of the pixel value has a value ranging from 0 to 255, these pixel pair values are selected and they are used as x and y co-ordinate values to locate the position in the 256X256 reference matrix. We find the value in the reference matrix at that position pointed by the pixel pair values and the secret digits that have to be embedded are selected and the closest position of the secret digit to that of the pointed values is selected and the pixel values are modified to the new values.

B. *Data extraction*

The embedded data can be extracted directly from the stego image by referencing the same Sudoku solution used in the embedding phase. To extract secret digits, stego image is partitioned into non overlapping blocks of 1×2 pixels using the same technique used in embedding phase. The pixel pair is chosen and the secret digit is simply obtained by referencing the reference matrix M at the position using the pixel values as the co-ordinate values.



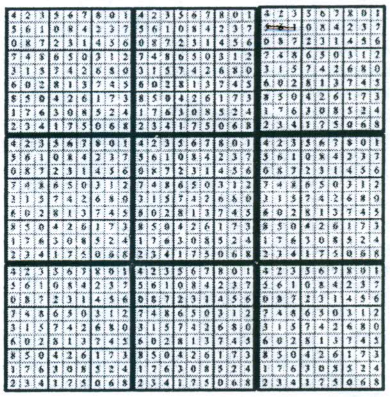
g_{i+1}

		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	...	252	253	254	255
	0	5	4	0	6	3	7	1	2	8	5	4	0	6	3	7	1	2	8	...	5	4	0	6
	1	8	2	7	1	4	5	0	6	3	8	2	7	1	4	5	0	6	3	...	8	2	7	1
	2	1	3	6	2	0	8	7	5	4	1	3	6	2	0	8	7	5	4	...	1	3	6	2
	3	2	1	3	0	8	6	5	4	7	2	1	3	0	8	6	5	4	7	...	2	1	3	0
	4	4	6	5	3	7	2	8	1	0	4	6	5	3	7	2	8	1	0	...	4	6	5	3
	5	7	0	8	4	5	1	2	3	6	7	0	8	4	5	1	2	3	6	...	7	0	8	4
	6	3	5	1	7	6	0	4	8	2	3	5	1	7	6	0	4	8	2	...	3	5	1	7
	7	6	7	2	8	1	4	3	0	5	6	7	2	8	1	4	3	0	5	...	6	7	2	8
	8	0	8	4	5	2	3	6	7	1	0	8	4	5	2	3	6	7	1	...	0	8	4	5
	9	5	4	0	6	3	7	1	2	8	5	4	0	6	3	7	1	2	8	...	5	4	0	6
g_i	10	8	2	7	1	4	5	0	6	3	8	2	7	1	4	5	0	6	3	...	8	2	7	1
	11	1	3	6	2	0	8	7	5	4	1	3	6	2	0	8	7	5	4	...	1	3	6	2
	12	2	1	3	0	8	6	5	4	7	2	1	3	0	8	6	5	4	7	...	2	1	3	0
	13	4	6	5	3	7	2	8	1	0	4	6	5	3	7	2	8	1	0	...	4	6	5	3
	14	7	0	8	4	5	1	2	3	6	7	0	8	4	5	1	2	3	6	...	7	0	8	4
	15	3	5	1	7	6	0	4	8	2	3	5	1	7	6	0	4	8	2	...	3	5	1	7
	16	6	7	2	8	1	4	3	0	5	6	7	2	8	1	4	3	0	5	...	6	7	2	8
	17	0	8	4	5	2	3	6	7	1	0	8	4	5	2	3	6	7	1	...	0	8	4	5
		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	252	5	4	0	6	3	7	1	2	8	5	4	0	6	3	7	1	2	8	...	5	4	0	6
	253	8	2	7	1	4	5	0	6	3	8	2	7	1	4	5	0	6	3	...	8	2	7	1
	254	1	3	6	2	0	8	7	5	4	1	3	6	2	0	8	7	5	4	...	1	3	6	2
	255	2	1	3	0	8	6	5	4	7	2	1	3	0	8	6	5	4	7	...	2	1	3	0

Fig 3. 256 X 256 Reference matrix

III. PROPOSED METHOD

Steganography using Sudoku is used to hide data or secret information onto an image using Sudoku solution. The image used in our proposed method is a 24 bit BMP image. Initially the data to be hidden is chosen which can be any digital media file such as text, image, audio, video etc. This digital media is converted to Base-9 for mapping them onto Sudoku solution. Sudoku solution is taken and every value in it is subtracted by '1'. This is done so as to ensure all values lie between 0 to 8 in Sudoku, for maintaining compatibility between input data which is in Base-9 format and the Sudoku values. This Sudoku solution is then expanded to 27 X 27 matrix called reference matrix (M) as shown in Fig 4.



4	2	3	5	6	7	8	0	1	4	2	3	5	6	7	8	0	1	4	2	3	5	6	7	8	0	1
5	6	1	0	8	4	2	3	7	5	6	1	0	8	4	2	3	7	5	6	1	0	8	4	2	3	7
0	8	7	2	3	1	4	5	6	0	8	7	2	3	1	4	5	6	0	8	7	2	3	1	4	5	6
7	4	8	6	5	0	3	1	2	7	4	8	6	5	0	3	1	2	7	4	8	6	5	0	3	1	2
3	1	5	7	4	2	6	8	0	3	1	5	7	4	2	6	8	0	3	1	5	7	4	2	6	8	0
6	0	2	8	1	3	7	4	5	6	0	2	8	1	3	7	4	5	6	0	2	8	1	3	7	4	5
8	5	0	4	2	6	1	7	3	8	5	0	4	2	6	1	7	3	8	5	0	4	2	6	1	7	3
1	7	6	3	0	8	5	2	4	1	7	6	3	0	8	5	2	4	1	7	6	3	0	8	5	2	4
2	3	4	1	7	5	0	6	8	2	3	4	1	7	5	0	6	8	2	3	4	1	7	5	0	6	8
4	2	3	5	6	7	8	0	1	4	2	3	5	6	7	8	0	1	4	2	3	5	6	7	8	0	1
5	6	1	0	8	4	2	3	7	5	6	1	0	8	4	2	3	7	5	6	1	0	8	4	2	3	7
0	8	7	2	3	1	4	5	6	0	8	7	2	3	1	4	5	6	0	8	7	2	3	1	4	5	6
7	4	8	6	5	0	3	1	2	7	4	8	6	5	0	3	1	2	7	4	8	6	5	0	3	1	2
3	1	5	7	4	2	6	8	0	3	1	5	7	4	2	6	8	0	3	1	5	7	4	2	6	8	0
6	0	2	8	1	3	7	4	5	6	0	2	8	1	3	7	4	5	6	0	2	8	1	3	7	4	5
8	5	0	4	2	6	1	7	3	8	5	0	4	2	6	1	7	3	8	5	0	4	2	6	1	7	3
1	7	6	3	0	8	5	2	4	1	7	6	3	0	8	5	2	4	1	7	6	3	0	8	5	2	4
2	3	4	1	7	5	0	6	8	2	3	4	1	7	5	0	6	8	2	3	4	1	7	5	0	6	8
4	2	3	5	6	7	8	0	1	4	2	3	5	6	7	8	0	1	4	2	3	5	6	7	8	0	1
5	6	1	0	8	4	2	3	7	5	6	1	0	8	4	2	3	7	5	6	1	0	8	4	2	3	7
0	8	7	2	3	1	4	5	6	0	8	7	2	3	1	4	5	6	0	8	7	2	3	1	4	5	6
7	4	8	6	5	0	3	1	2	7	4	8	6	5	0	3	1	2	7	4	8	6	5	0	3	1	2
3	1	5	7	4	2	6	8	0	3	1	5	7	4	2	6	8	0	3	1	5	7	4	2	6	8	0
6	0	2	8	1	3	7	4	5	6	0	2	8	1	3	7	4	5	6	0	2	8	1	3	7	4	5
8	5	0	4	2	6	1	7	3	8	5	0	4	2	6	1	7	3	8	5	0	4	2	6	1	7	3
1	7	6	3	0	8	5	2	4	1	7	6	3	0	8	5	2	4	1	7	6	3	0	8	5	2	4
2	3	4	1	7	5	0	6	8	2	3	4	1	7	5	0	6	8	2	3	4	1	7	5	0	6	8

Fig 4. 27 X 27 Reference matrix

A. *Data embedding*

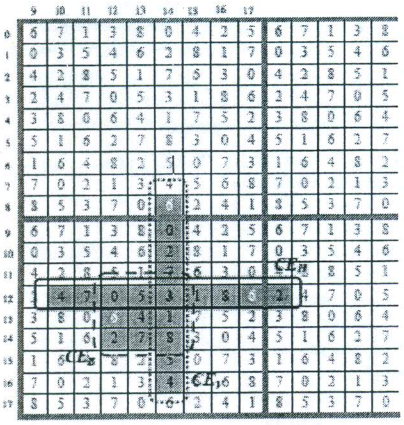
Any image onto which secret information has to be embedded is chosen. Each pixel of this image is extracted and two components of pixel: Red (R) and Green (G) colour are chosen for embedding. Each colour component is an 8 bit binary number. The 8 bits represent numbers ranging from 0 to 255. This value is converted to a value between 0 to 8 using the following formulae:

$$R = R \% 9, G = G \% 9$$

To this value of R and G, 9 are added for ensuring its value is located at the center of the reference matrix. Then R and G are chosen as X-axis and Y-axis components of reference matrix M, forming pair (g_i, g_{i+1}), where g_i = R and g_{i+1} = G. Then three candidate elements are chosen called Horizontal (CE_H), Vertical (CE_V) and Boxed (CE_B) as shown in figure 5. Here CE_H is shown by thick line, CE_V is shown by dotted line and CE_B is shown by dashed lines in fig 5. These candidate elements are determined as follows:

```
S1: (i.e. select the candidate elements for $CE_H$)
        If $g_{i+1} > 3$ and $g_{i+1} < 252$,
        then $g_i=g_i\%9+9$ , $g_{i+1}\%9+9$ and
                $CE_H = \{M(g_i, g_{i+1} - 4), M(g_i, g_{i+1} - 3),$
                $M(g_i, g_{i+1} - 2), M(g_i, g_{i+1} - 1),$
                $M(g_i, g_{i+1}), M(g_i, g_{i+1} + 1), M(g_i, g_{i+1} +2),$
                $M(g_i, g_{i+1} +3), M(g_i, g_{i+1} +4)\};$
        Else If $g_{i+1} \leq 3$,
        then $g_i=g_i\%9+9$ , $g_{i+1}\%9+9$ and
                $CE_H = \{M(g_i, 9), M(g_i, 10),$
```

```
            $M(g_i, 11), M(g_i,12), M(g_i,13), M(g_i,14),$
            $M(g_i,15),M(g_i,16), M(g_i,17)\};$
    Else If $g_{i+1} \geq 252$,
    then $g_i=g_i\%9+9$ and $g_{i+1}\%9+9$ and
            $CE_H = \{M(g_i, 4), M(g_i, 5), M(g_i, 6),$
            $M(g_i, 7), M(g_i, 8), M(g_i, 9), M(g_i, 10),$
            $M(g_i, 11), M(g_i, 12)\}$.
S2: (i.e. select the candidate elements for $CE_V$ )
    If $g_i > 3$ and $g_i < 252$,
    then $g_i=g_i\%9+9$ , $g_{i+1}\%9+9$ and
            $CE_V = \{M(g_i - 4, g_{i+1}), M(g_i -3 , g_{i+1}),$
            $M(g_i -2, g_{i+1}), M(g_i -1, g_{i+1}), M(g_i, g_{i+1}),$
            $M(gi+1, g_{i+1}), M(gi+2, g_{i+1}),$
            $M(gi +3, g_{i+1}),M(gi +4, g_{i+1})\};$
    Else If $g_i \leq 3$,
    then $g_i=g_i\%9+9$ , $g_{i+1}\%9+9$ and
            $CE_V = \{M(9, g_{i+1}), M(10, g_{i+1}),$
            $M(11, g_{i+1}), M(12, g_{i+1}), M(13, g_{i+1}),$
            $M(14, g_{i+1}), M(15, g_{i+1}), M(16, g_{i+1}),$
            $M(17, g_{i+1})\};$
    Else If $g_i \geq 252$,
    then $g_i=g_i\%9+9$ , $g_{i+1}\%9+9$ and
            $CE_V = \{M(4, g_{i+1}), M(5, g_{i+1}), M(6, g_{i+1}),$
            $M(7, g_{i+1}), M(8, g_{i+1}), M(9, g_{i+1}),$
            $M(10, g_{i+1}), M(11, g_{i+1}), M(12, g_{i+1})\}$.
S3: (i.e. select the candidate elements for $CE_B$)
    If $g_i < 252$ and $g_{i+1} < 255$,
    then compute $x_b = \lfloor g_i /3 \rfloor \times 3$,
            $y_b = \lfloor g_{i+1} /3 \rfloor \times 3$, and
            $CE_B = \{M(x_b, y_b), M(x_b, y_b + 1),$
            $M(x_b, y_b + 2), M(x_b + 1, y_b),$
            $M(x_b + 1, y_b +1), M(x_b +1, y_b +2),$
            $M(x_b +2, y_b), M(x_b +2, y_b + 1),$
            $M(x_b + 2, y_b + 2)\}$
    Else $CE_B$ = <EMPTY>.
```



	9	10	11	12	13	14	15	16	17					
0	6	7	1	3	8	0	4	2	5	6	7	1	3	8
1	0	3	5	4	6	2	8	1	7	0	3	5	4	6
2	4	2	8	5	1	7	6	3	0	4	2	8	5	1
3	2	4	7	0	5	3	1	8	6	2	4	7	0	5
4	3	8	0	6	4	1	7	5	2	3	8	0	6	4
5	5	1	6	2	7	8	3	0	4	5	1	6	2	7
6	1	6	4	8	2	5	0	7	3	1	6	4	8	2
7	7	0	2	1	3	4	5	6	8	7	0	2	1	3
8	8	5	3	7	0	6	2	4	1	8	5	3	7	0
9	6	7	1	3	8	0	4	2	5	6	7	1	3	8
10	0	3	5	4	6	2	8	1	7	0	3	5	4	6
11	4	2	8	5	1	7	6	3	0	CE_H	2	8	5	1
12	2	4	7	0	5	3	1	8	6	2	4	7	0	5
13	3	8	0	6	4	1	7	5	2	3	8	0	6	4
14	5	1	6	2	7	8	3	0	4	5	1	6	2	7
15	1	6	CE_B	8	2	5	0	7	3	1	6	4	8	2
16	7	0	2	1	3	4	CE_V	6	8	7	0	2	1	3
17	8	5	3	7	0	6	2	4	1	8	5	3	7	0

Fig 5. Illustration of CE_H (solid line), CE_V (dotted line), CE_B (dashed line)

Then location of each digit of secret information (which is in base-9 format) to be embedded is found with respect to R and G. According to an input secret digit S_i, three candidate elements $M(x_H, y_H)$, $M(x_V , y_V)$, and $M(x_B, y_B)$ are found from CE_H, CE_V , and CE_B, respectively. In other words, if g_i, and g_{i+1} are smaller than 255, then the found candidate elements satisfy $M(x_H, y_H) = M(x_V , y_V) = M(x_B, y_B) = S_i$; otherwise, $M(x_H, y_H) = M(x_V , y_V) = S_i$. The cover pixel pair (g_i, g_{i+1}) is modified as (g_i^1, g_{i+1}^1) by a minimum distortion candidate element $M(x_{min}, y_{min})$ which is selected by using Manhattan distance formula:

$$M(x_{min}, y_{min}) = \min_j =H,V,B\{| g_i - x_j | + | g_{i+1} - y_j|\}$$

Thus, the cover pixel pair (g_i, g_{i+1}) is modified as $(g_i^1 = x_{min}, g_{i+1}^1 = y_{mn})$ to conceal the secret digit S_i with small distortion. This small distortion does not affect the image much and no physical difference can be identified of this image with respect to original image. After performing the above replacement for all digits of secret information the embedding phase is complete.

The main requirement for this project is that the Sudoku solution has to be same at both sender and receiver end. The size of secret information is embedded initially on the cover image so that receiver can determine how many pixels to scan and retrieve data. Since digital media can be any type of data such as text, image, audio etc; it is necessary for the receiver to be aware of it. This is done by embedding them after embedding secret digits on the cover image itself. This is done by embedding the size of extension followed by the extension itself. The overall description of embedding phase is as shown in Fig 6.

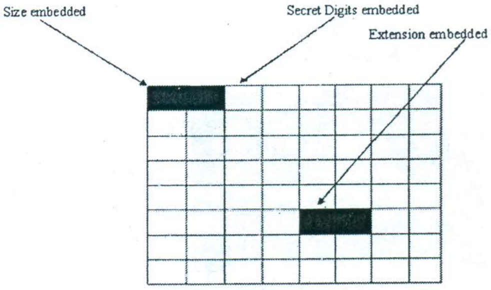


Fig 6. General format of embedding data

As shown in figure above size is embedded in first 9 pixels of cover image (represented by black colored boxes). Then secret digits are embedded till the embedded size (represented by white boxes). This is then followed by extension size and extension itself (represented by black colored boxes). The rest of the pixels in the image that follow are retained as it is.

B. Data extraction

The embedded secret digits can be exactly extracted from the received stego image with the same Sudoku solution used in the embedding phase. In this phase first each pixel is extracted from the stego image. Then from this red (R) and green (G) components of each pixel are used (similar to embedding phase). Their pixel values are taken and converted to a value between 0 to 8 using the following formulae:

$$R = R \% 9, G = G \% 9$$

The Sudoku solution which is shared between the user and the receiver is converted into a reference matrix (9x9 matrix) by subtracting 1 from the Sudoku solution, similar to the one shown in fig 2.

The R and G are chosen as X-axis and Y-axis components of Sudoku solution, forming pair (g_i, g_{i+1}), where g_i = R and g_{i+1} = G. The value at position ($g_{i.}$ g_{i+1}) is the required secret digit. This process is done for all pixel and data is extracted. The obtained secret digit (which is in base-9) is converted to base-2. This completes extraction phase. In the previously proposed method the reference matrix required for extraction is 256X256 matrix but here we make use of a 9X9 reference matrix for extraction. The main advantage of this method is that the distortion of each pixel is in the range 0f '-4' to '+4' based on the candidate element, thus resulting in minimum distortion.

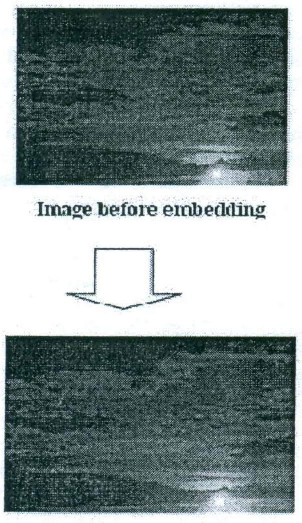


Image before embedding

Image after embedding

Fig 7. Sample image before and after embedding data

IV. CONCLUSION

Steganography is the science of secret data delivery. In this paper, a novel information hiding method is proposed to significantly improve embedding capacity by using Sudoku. Because the property of Sudoku (i.e. a large number of possible solutions), the proposed method is secure. The number of possible Sudoku solutions is 6.71×10^{21} [5]. In Chang et al.'s method, a reference matrix of 256x256 was used which meant that we needed a two dimensional array of 256x256 when we implement it using some programming language. But this is not feasible, as this will cause a problem of insufficient memory. A normal computer memory cannot accommodate 256x256 memory bytes to be allocated at once. To solve this problem we are making use of a 27x27 matrix and are using simple modulo operation along with addition and subtraction. 27x27 can be easily accommodated in the memory. In our method we use a 27X27 reference matrix during embedding phase and a 9X9 reference matrix during extraction phase. Only 1 pixel is enough to embed the data since we use color images (we use the red and the green component of each pixel). This method of data hiding can be used anywhere like for military applications etc., since the quality of the image is not affected much, before and after embedding as shown in Fig. 7.

V. OTHER RECOMMENDATION

There could be lot of scope for further enhancement to this project, some of which are listed here.

1. In this proposed method we have modified only the red and the green components of the image. The blue component of the image can also be modified. In that way a higher embedding capacity can be obtained.
2. Two or more digital media files (input files) can be embedded in a single image file.
3. This method also can be implemented for GIF or PNG images which are lossless images.

ACKNOWLEDGEMENT

The authors acknowledge and thank the support of Professor B.I. Khodanpur, Head Department of Computer Science and Engineering, R. V. College of Engineering for his constant support, suggestions and guidance.

REFERENCES

[1] C.-C. Chang, T. D. Kieu, and Y.-C. Chou. High capacity data hiding for grayscale images. In *Proceedings of the First International Conference on Ubiquitous Information Management and Communication*, pages 139–148. Seoul, Korea, February 2007.

[2] C.-C. Chang and C.-Y. Lin. Reversible steganography for vq-compressed images using side matching and relocation. *IEEE Transactions on Information Forensics and Security*, 1(4):493–501, December 2006.

[3] Y.-T. Wu and F. Y. Shih. Digital watermarking based on chaotic map and reference register. *Pattern Recognition*, 40(12):3754–3763, December 2007.

[4] C.C. Chang, Y.C. Chou and T.D. Kieu, An Infonnation Hiding Scheme Using Sudoku, Proceedings of the Third International Conference on Innovative Computing, Information and Control (ICICIC2008), June 2008.

[5] B. Felgenhauer and F. Jarvis. Mathematics of Sudoku I, Mathematical Spectrum, vol. 39, no. 1, pp. 15-22, 2006.