A Novel Steganography with Dynamic Start Point for Encoding using Sudoku

Md. Habibur Rahaman, Dr. Sajjad Waheed

Abstract— 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. A new, simple, approach for active steganography is proposed in this paper that can successfully resist recent blind steganalysis methods, in addition to surviving distortion constrained attacks. The proposed method, keep the same technique to embed the secret key from Arnab Kumar Maji's using 8X8 Sudoku puzzle. The security have increased using dynamic encoding start point for both message and secret key encoding.

Index Terms— - Steganography, Embedding, data hiding, RGB Image, pixels, secret data, Secret information, Extraction.

1 Introduction

THE word 'Steganography' was derived from Greek words 'stegos', meaning 'cover'; and 'graphia', meaning 'writing'. This is a way of hiding any message within an ordinary message. The extraction of the message can be done when it reaches destination. In modern digital steganography, data secrecy is achieved through various secure algorithms. For an image, steganography algorithms are used to replace the old pixels with new pixels those have very small distortion. 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 stego analysis. So security in steganography can be achieved using different embedding schemes [4-5]. The important factors needed to consider when designing a new information hiding scheme are: embedding capacity (i.e. the number of secret bits can be embedded into one cover image pixel), visual quality of stego images [6] (i.e. image distortion); amount of data sent (i.e. compression) and secure exchange of data (i.e. Encryption). Desirably, one would want to achieve high embedding capacity, good visual quality, and more data to get embedded and high security. 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.

The simplest approach in hiding data within an image file is called least significant bit (LSB) insertion [4]. In this method, take the binary representation of the hidden_data and overwrite the LSB of each byte within the cover_image. If 24-bit colour image has used, the amount of change is minimal and indiscernible to the human eye. As an example, suppose that, there are three adjacent pixels (nine bytes) with the following RGB encoding:

10010101 00001101 11001001
10010110 00001111 11001010
10011111 00010000 11001011

Now anyone want to hide the following nine bits of data (the hidden data is usually compressed prior to being hidden): 101101101. If these nine bits overlayed over the LSB of the nine bytes above, result will be (where bits in bold have been changed):

10010101 00001100 11001001
10010111 00001110 11001011
10011111 00010000 11001011

One of the potential problems of this type of method is that any intruder can easily modify the cover_image. Then it is impossible for the receiver to extract the hidden message int he cover_image. So, detecting whether a cover_image is modified during transmission is very much essential. 'Sudoku' is a popular Japanese puzzle game. Arnab Kumar Maji's [9] used 8X8 sudoku for secret key, so that modification was detected.

2 Literature Servey

Initial work on Steganography using Sudoku puzzle was done by Chang, Chou, and Kieu [3]. The basic idea in the method is to use a Sudoku puzzle to generate a reference matrix (M) and alter the values at selected pixels in cover image according to values represented in reference matrix. For an 8-bit cover image, the size of reference matrix is 256-by-256. After that Arnab Kumar Maji's [9] proposed a method that was more secured and need less computations that the previous one.

2.1 Maji et all method [3]

1. Generate an 8-by-8 Sudoku for detecting modifications in the cover image and an 18-by-18 Sudoku reference matrix is used as key, which is lesser in size than that of each of the previous methods. The 24-bit colored image is used in this proposed method. Initially the cover_image is divided into 64 blocks. In each block, several groups of three pixels each are created. Then, extract the values of B (Blue) components of each pixel, which is represented by eight bits. Now take an 8-

International Journal of Scientific & Engineering Research, Volume 8, Issue 5, May-2017
ISSN 2229-5518

by-8 Sudoku puzzle. Subtract 1 from each cell. Then the value in each cell be ranging from 0 - 7, each of which can be represented by three bits. For each three pixels group of a block, insert the values in the LSB of B (Blue) components. Thus only one bit is changed.

2. After that, an 18-by-18 reference matrix is generated from a 9-by-9 Sudoku puzzle. The reference matrix is created by replicating 9-by-9 Sudoku solution in a square form. The values of the reference matrix lie between 0 through 8 as 1 is subtracted from each value. The reference matrix behaves as a reference look-up table for embedding. A sample 18-by-18 Sudoku reference matrix is generated.

3. Convert the decimal (ASCII code) to base-9. Add padding bits (0's) in front of the base-9 number, if necessary, so that the number is a 3-digit number. Finding out the base-9 values in reference matrix: For each digit of the three-digit base-9 character code Read pixel (h, f) /*h and f are random location of pixel*/

X = (R%6) + 6,

Y = (G%6) + 6.

Locate the cell (X, Y) in the reference matrix. X can be considered as row number and Y as column number. Now consider nine cells in the same row, keeping the cell (X,Y) in the middle of it and store them in Cr, i.e., consider only four left-most and four right-most cells,

Surrounding (X, Y). Similarly, choose nine cells in the same column, keeping the cell (X,Y) in the middle of them, and store them in Cc. Select the minigrid, where the cell (X,Y) belongs to and store them in Cm.

If (Xi, Yi) = base-9 digit, then Locate (Xr, Yr), (Xc, Yc), and (Xm, Ym) in Cr, Cc, and Cm of the reference matrix, respectively. Calculate deviation of (Xr, Yr), (Xc, Yc), (Xm, Ym) from (X, Y).

Select the cell from the above three candidate elements which has minimum deviation from (X, Y).

(iii) Updating the R and G values of each pixel: Suppose, (Xc, Yc) has minimum deviation from (X, Y), then The pixel at (h, f) has new data-embedded R-G-B values as:

R = R - (X - Xc)

G = G - (Y - Yc)

B holds the 8-by-8 Sudoku value embedded previously.

4. For extracting the hidden message from the stego_image, transfer the instance of 9 X 9 Sudoku puzzle. Then after receiving it the receiver solves this puzzle and an 18 X 18

Reference matrix is generated. Then,

For each bit in a three-pixel group, compute the following:

X = (R%6) + 6,

Y = (G%6) + 6.

Find the value present at (X, Y) from the reference matrix.

End for

Concatenate the three values from the three pixels.

Convert the result to base-10. This is an ASCII value of the hidden character message.

Get the character equivalent to the decrypted ASCII code.

But there are several limitations or drawbacks of this method.

(i) The secret key embedding procedure is still less secure and can't resist recent blind steganalysis methods.

(ii) If somehow anyone can detect the secret key of any block by reading the LSB from three pixel group the message can be retrieved easily for that.

3 Proposed Method

8-by-8 Sudoku matrix has used for detecting modifications in the cover image. The image used in this proposed method is a 24-bit colored image. Initially, the cover picture is taken and an 8-by-8 Sudoku is embedded onto it using the LSB embedding technique that is slightly different from *Arnab Kumar Maji's* method [9].

3.1 Embedding an 8-by-8 Sudoku matrix in cover_image

(i) Block preparation:

The cover_image is divided into 64 blocks. In each block, several groups of three pixels each are created. Then, extract the values of B (Blue) components of each pixel, which is represented by eight bits. Now take an 8-by-8 Sudoku puzzle. Subtract 1 from each cell. Then the value in each cell be ranging from 0 - 7, each of which can be represented by three bits. An instance of 8-by-8 Sudoku puzzle and its possible solution are shown in Figure 1.

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

Fig. 1. An instance of 8-by-8 Sudoku puzzle.

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

Fig. 2. A solution of the Sudoku instance.

International Journal of Scientific & Engineering Research, Volume 8, Issue 5, May-2017
ISSN 2229-5518

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

Fig. 3. Sudoku tile matrix (subtract 1 from each cell of the Sudoku solution).

The embedding technique of this Sudoku values into image is done with following manner:

Take a value from 8-by-8 Sudoku,

For ***First*** and ***Second*** pixel: Change the last (which is 8th) bit of B (Blue) component.

For ***Third*** pixel: If the Sudoku block value is *even* than the last (which is 8th) bit of B(Blue) component will change and if the value is *Odd* then change the (last-1)th (which is 7th) bit of B(Blue) component.

It will prevent the blind reading of pixels from staganalyst.

The start point for embedding both the *secret key* and *message* is chosen from the following manner:

start_point = (image_block_height X image_block_width) / sudoku_value_for_this_block-encoding_direction

For example, take an image of 1000 X 800 pixel, add the sudoku tile matrix into this image shown in figure 3-(d). So each block will be 125 X 100. The sudoku value for second block is 5. Now calculating the encoding_direction, start_point as follows:

start_point = (125 X 100)/5=2500, so encoding starts from pixel no 2500 and continue with *Left to Right* method. So the binary of 5 is 101 (add padding 0 at first to make 3 digit binary number)

First 1 is replaced with the LSB of B (blue) component of pixel no 2500.

Second 0 is replaced with the LSB of B (blue) component of pixel no 2501 and

Third 1, check the Sudoku block value is *even* or *odd*, as 5 is odd number so the 7th bit of B (blue) component of pixel no 2502 has changed, it will continue until all pixel group are finished to encode.

3.2 Embedding hidden message

All tables and figures will be processed as images. You need to embed the images in the paper itself. Please don't send the images as separate files.

3.3 Checking the integrity of the cover_image

For checking whether the cover_image has been modified or not, first sender need to send an 8-by-8 Sudoku instance to the receiver. Receiver solves this puzzle and stores it. Then the Receiver divides the cover_image into 64 blocks and in each of the blocks, a group of three pixels be prepared. Then the last bit of first two pixel are taken and the 7th or 8th bit are taken (based on the Sudoku block value, that are discussed previously) from third pixel among three pixel group is taken and the equivalent decimal values are kept in the 8-by-8 matrix. If this matrix match with the already solved Sudoku puzzle, then the integrity has been maintained; otherwise, the image has been modified. Hence the attack is detected. Here the third pixel save the blind attack for security code, because it will be 8th bit or 7th bit multiply with the image total pixel divided by 3.

3.4 Hidden message extraction

For extracting the hidden message from the stego_image, first sender need to send the 8-by-8 Sudoku instance to the receiver. Receiver solves the puzzle and the puzzle are kept into the stego_image, and based on the value of each block he find out start_point (that has been discussed in encoding part) for each 64 block. And from each three pixel group he will get a 7 bit binary number and then convert the binary to decimal that is the ASCII of the hidden character.

3.5 Sending of Sudoku instances

The sender of the secret message needs to send an 8-by-8 Sudoku instance to the receiver.

The whole process (i.e., the proposed Steganographic scheme) is shown in Figure 4.

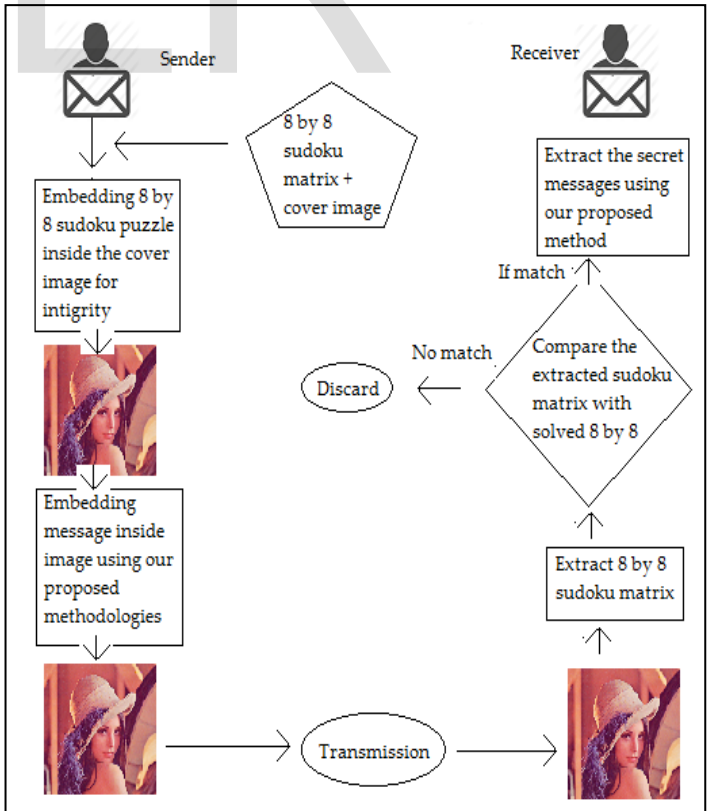

Fig. 4. Proposed steganographic scheme.

International Journal of Scientific & Engineering Research, Volume 8, Issue 5, May-2017
ISSN 2229-5518

4 Result Obtained

The quality of stego_image is evaluated using histogram comparison in MAT Lab and embedding capacity in terms of characters. A sample result has been shown as follows. After performing histogram analysis, it can easily be found that there are very less distortion in the cover_image. The embedding capacity for the image shown in Figure 7 is calculated in the following manner:

Sample image size = 462 KB.
Number of pixels in the image = 262144.
Number of pixels required for hiding one character = 3.
Therefore, 262144/3=87381 characters.

Thus, this method can hide approximately 85.33 KB secret messages in the image shown in Figure 7. Leena image has used for encoding. Compare the two image before encoding and after encoding:

Fig. 5. A sample cover_image.

Fig. 6. Stego_image after embedding the sample cover image shown in Figure 5.

Fig. 7. Histogram of the cover_image, shown in Figure 5.

Fig. 8. Histogram of the stego_image, shown in Figure 6.

It is very hard to find the change between the two images. And also there is less distortion in histogram comparison.

4.1 Comparison with existing method

The Chang et al.'s method [3] has used a reference matrix of size 256-by-256, whereas in the Shetty et al.'s method [7], a 27-by-27 reference matrix has been used. The achievements:

a. The reference matrix has eliminated to avoid the complex computation. So less computation need less time to encode and decode.

b. The data encoding start point has kept dynamic, so that it will resist the blind hacking.

c. Moreover, a separate 8-by-8 Sudoku matrix has embedded into the blue components of each pixel in each block, which provides an additional layer of security, so that any modification of the image was detected.

d. The image quality is good enough than the previous method.

5 Conclusion

In this method, a Steganographic scheme has proposed, that have dynamic starting point for each 64 block of an image, also embedding an 8-by-8 Sudoku, for checking whether the cover image has been modified (or not). If somehow the cover image gets modified, it can easily be detected as the 8-by-8 Sudoku matrix has already embedded inside it. It can also prevent any modification, as each Sudoku puzzle matrix should have values 0 through 8 only once in the same row, in the same column, and in a minigrid. The reference matrix has eliminated for encoding That is why, less computation is involved in this method whereas the entire earlier existing methods used 256 X 256 or 27 X 27 reference matrix. It can be claimed that the proposed scheme is more secured and robust with less computation.

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] http://www.en.wikipedia.org/wiki/Pentomino

[3] Roshan Shetty B R, Rohith J, Mukund V, Rohan Honwade, Shanta Rangaswamy. Steganography Using Sudoku Puzzle. 2009 International Conference on Advances in Recent Technologies in Communication and Computing,pages 623-626

International Journal of Scientific & Engineering Research, Volume 8, Issue 5, May-2017
ISSN 2229-5518

[4] http://www.en.wikipedia.org/wiki/Tetris
[5] Y.-T. Wu and F. Y. Shih. Digital watermarking based on chaotic map and reference register. Pattern Recognition, 40(12):3754–3763, December 2007.
[6] C.C. Chang, Y.C. Chou and T.D. Kieu, An Information Hiding Scheme Using Sudoku, Proceedings of the Third International Conference on Innovative Computing, Information and Control (ICICIC2008), June 2008.
[7] C. C. Chang, Y. C. Chou, and T. D. Kieu, "An Information Hiding Scheme Using Sudoku", in Proc. Third International Conference on Innovative Computing, Information and Control (ICICIC2008), June 2008.
[8] B. R. Shetty, J. Rohith, V. Mukund, and R. Honwade, "Steganography using Sudoku Puzzle", in Proc. International Conference on Advances in Recent Technologies in Communication and Computing, pp. 623-626, 2009.
[9] Arnab Kumar Maji, Rajat Kumar Pal, and Sudipta Roy, "A Novel Steganographic Scheme using Sudoku", in 2013 International Conference on Electrical Information and Communication Technology (EICT), August 2013.
[10] Pasumarthy Sarada1, Dr.Ch.BalaSwamy, "Improving Image Data Hiding Capacity Scheme using Sudoku Puzzle in Color Images", in International Journal of Engineering Research and Applications (IJERA), May-Jun 2012.
[11] Suman Chakraborty, Prof. Samir K. Bandyopadhyay, "Steganography Method Based On Data Embedding By Sudoku Solution Matrix", in (Online): 2319 – 6734, ISSN (Print): 2319 – 6726 www.ijesi.org Volume 2 Issue 7‖ July. 2013 ‖ PP.36-42, July. 2013.