

# Robust Steganography Using LSB-XOR and Image Sharing

Chandranath Adak

Department of Computer Science and Engineering  
University of Kalyani  
Kalyani – 741235 , India  
adak32@gmail.com

**Abstract**—Hiding and securing the secret digital information and data that are transmitted over the internet is of widespread and most challenging interest. This paper presents a new idea of robust steganography using bitwise-XOR operation between stego-key-image-pixel LSB (Least Significant Bit) value and secret message- character ASCII-binary value (or, secret image-pixel value). The stego-key-image is shared in dual-layer using odd-even position of each pixel to make the system robust. Due to image sharing, the detection can only be done with all the image shares.

**Keywords**—Image Sharing; Robustness; Steganography

## I. INTRODUCTION

Steganography has gained significance over years for its art of writing hidden messages in such a way, that no one, apart from the sender and intended recipient, suspects the existence of the message, it is a form of security through obscurity[1]. The term *steganography* is derived from two Greek words *steganos*, meaning “covered”, and *graphein*, meaning “to write”. The first recorded use of the term was in 1499 by Johannes Trithemius in his *Steganographia*, a treatise on cryptography and steganography disguised as a book on magic[2]. The history of steganography can be traced back to around 440 B.C., where the Greek historian Herodotus described in his writing about two events : one used wax to cover secret messages, and the other used shaved heads[3]. Modern steganography entered the world in 1985 with two engineers, Barrie Morgan and Mike Barney[4].Digital image is the most popular carrier in the study of steganography. Steganography differs from cryptography in the sense that where cryptography focuses on keeping the contents of a message secret, steganography focuses on keeping the existence of a message secret[5-11].

There has been many approaches in steganography, which includes StegHide[12], OutGuess[13], model based steganography[14], perturbed quantization[15] and statistical restoration[16], but the most traditional approach is hiding data in LSB[17] of spatial or transform domain coefficient.

Naor & Shamir[18] demonstrated a visual secret sharing scheme, where an image was broken up into  $n$  shares so that only someone with all  $n$  shares could decrypt the image, while

any  $n-1$  shares revealed no information about the original image. In the prescribed cryptosystem, the stego-key-image is broken into two slides.

Here, we are presenting a stego-system using digital image as a carrier (Stego Key), and LSB-XOR technique with image sharing concept of visual cryptography.

## II. STEPS OF PROPOSED METHOD

### A. Stego-Key Generation

An arbitrary grey image (Stego-Key-image) is chosen for stego-key generation. The grey value of each pixel is extracted from this image and using odd-even position of those pixels two shared image ( $S_{\text{Key img1}}$  and  $S_{\text{Key img2}}$ ) is created. The grey values of these shared images are converted into equivalent eight-bit binary numbers and the pool of these LSBs is the stego-key values.

### B. Embedding Algorithm

The secret message is either ‘textual message’ or ‘image message’ ; for textual-message the ASCII (American Standard Code for Information Interchange) value of each character and for image-message the grey value of each pixel is used. The ASCII values (or, grey values) are treated as decimal numbers and converted into equivalent eight-bit binary numbers.

These eight-bit binary values are stored by its odd-even position to make  $S_{\text{Msg1}}$  (using only odd positioned four bits) and  $S_{\text{Msg2}}$  (using only even positioned four bits).

The  $S_{\text{Msg1}}$  and  $S_{\text{Msg2}}$ ’s bits are bitwise-XORed with LSBs of  $S_{\text{Key img1}}$  and  $S_{\text{Key img2}}$  respectively. After this bitwise XOR operation, the new binary grey values are converted into equivalent decimal grey values. These grey values are stored in odd-even position to form Stego-Image. This Stego-Image is transmitted through the channel.

### C. Detector Algorithm

The Stego-Image’s pixel values are treated as decimal numbers and converted into eight bit binary numbers. The Stego-Image is shared in two parts ( $S_{\text{Img1}}$  and  $S_{\text{Img2}}$ ) with odd-even positioned pixel grey values.

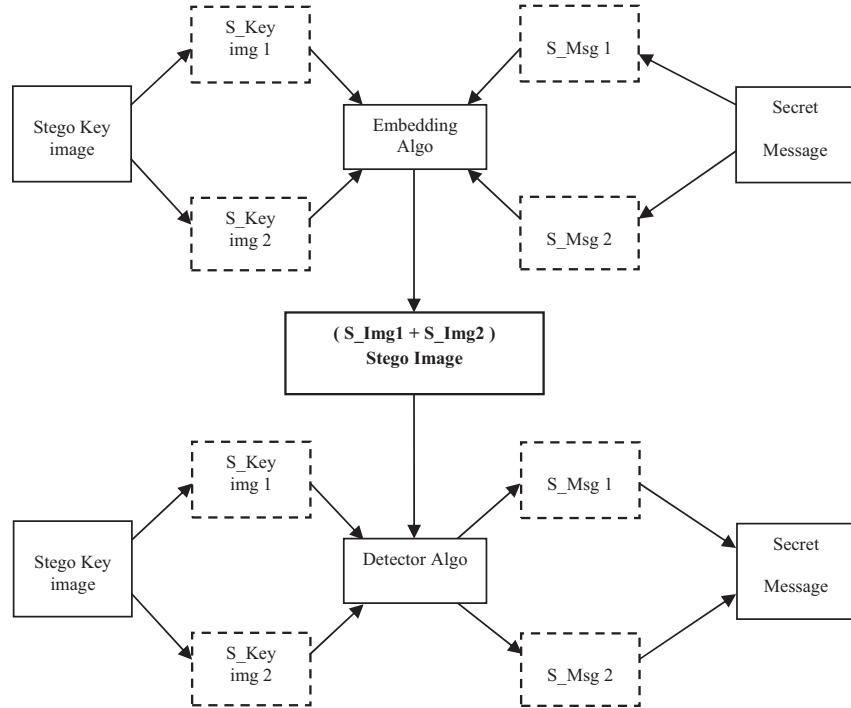


fig. 1 : Model of the proposed method

The S\_Img1 and S\_Img2's LSBs are bitwise XORed with LSBs of S\_Key img1 and S\_Key img2 respectively.

The bitwise-XOR operated LSBs are stored by odd-even position and taken as pool of eight bit binary numbers. These eight-bit binary numbers are converted into equivalent decimal numbers and seemed as either ASCII values for textual-message or grey values for image-message. The secret message is detected with these values.

The model of this proposed method is shown in fig.1 .

### III. IMPLEMENTATION

The following is the illustration of the proposed methodology with an example.

Let, the Secret Message is textual message and it contains a single letter "I". The secret Stego-Key-image is the Lena image.

In the embedding process, at sender side, the odd and even positioned Stego-Key-image pixels form S\_Key img1 and S\_Key-img2 respectively by image sharing in dual part as shown in table 1.

Pixel Co-ordinate of Stego-Key-Image	Stego-Key-image-pixel grey values	Equivalent 8-bit binary
(0,0)	162	10100010
(0,1)	161	10100001
(0,2)	158	10011110
(0,3)	156	10011100
(0,4)	156	10011100
(0,5)	153	10011001
(0,6)	154	10011010
(0,7)	161	10100001
(0,8)	168	10101000
(0,9)	173	10101101
.	.	.
.	.	.
.	.	.

Table 1.(a)

Pixel Co-ordinate of S_Key img 1	Pixel Co-ordinate of Stego-Key-image	S_Key img 1-pixel grey values	Equivalent 8-bit binary (B_SKey1)
(0,0)	(0,0)	162	10100010
(0,1)	(0,2)	158	10011110
(0,2)	(0,4)	156	10011100
(0,3)	(0,6)	154	10011010
(0,4)	(0,8)	168	10101000
.	.	.	.
.	.	.	.
.	.	.	.

Table 1.(b)

Pixel Co-ordinate of S_Key img 2	Pixel Co-ordinate of Stego-Key-image	S_Key img 2-pixel grey values	Equivalent 8-bit binary (B_SKey2)
(0,0)	(0,1)	161	10100001
(0,1)	(0,3)	156	10011100
(0,2)	(0,5)	153	10011001
(0,3)	(0,7)	161	10100001
(0,4)	(0,9)	173	10101101
.	.	.	.
.	.	.	.
.	.	.	.

Table 1.(c)

Table 1.(a),(b),(c) show Stego-Key-image , S\_Key-img1 and S\_Key-img2 pixel grey values and their equivalent 8-bit binary. The LSBs are shown in bold format.

The Secret Message (textual) is “I”, its ASCII value is ‘73’, which is treated as decimal number and converted into eight bit binary number.  $(73)_{10} = (01001001)_2$ . This binary number is stored in the reverse way and shared in two parts with respect to odd-even position as shown in fig.2 .

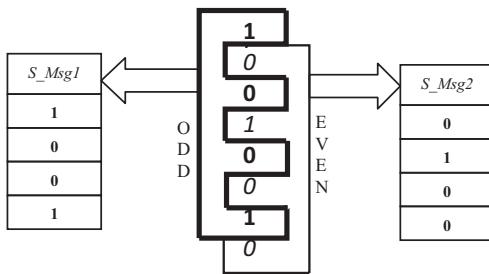


fig. 2 : Secret-Message partitioning

B\_SKey1 and B\_SKey2 are the equivalent eight-bit binary of S\_Key-img1 and S\_Key-img2 pixel grey values respectively. The LSBs of B\_SKey1 and B\_SKey2 are bitwise-XORED with S\_Msg1 and S\_Msg2 respectively as table 2 .

LSB of B_SKey1 (i1)	S_Msg1 (sm1)	i1 = II ^ sm1
0	1	1
0	0	0
0	0	0
0	1	1

Table 2.(a)

LSB of B_SKey2 (i2)	S_Msg2 (sm2)	i2 = I2 ^ sm2
1	0	1
0	1	1
1	0	1
1	0	1

Table 2.(b)

Table 2.(a) shows the bitwise XOR operation between LSB of B\_SKey1 and S\_Msg1 ; Table 2.(b) shows the bitwise XOR operation between LSB of B\_SKey2 and S\_Msg2 .

B_SKey1	i1	S_Img1 pixel binary grey value (replacing LSB of B_SKey1 with i1)	S_Img1 pixel decimal grey value (after replacement)
10100010	1	10100011	163
10011110	0	10011110	158
10011100	0	10011100	156
10011010	1	10011011	155

Table 3.(a)

B_SKey2	i2	S_Img2 pixel binary grey value (replacing LSB of B_SKey2 with i2)	S_Img2 pixel decimal grey value (after replacement)
10100001	1	10100011	161
10011100	1	10011111	157
10011001	1	10011101	153
10100001	1	10011011	161

Table 3.(b)

Pixel Co-ordinate of Stego-Image	Stego-Image-pixel binary grey values	Equivalent decimal grey values
(0,0)	10100011	163
(0,1)	10100011	161
(0,2)	10011110	158
(0,3)	10011111	157
(0,4)	10011100	156
(0,5)	10011101	153
(0,6)	10011011	155
(0,7)	10011011	161
(0,8)	10101000	168
(0,9)	10101101	173
.	.	.
.	.	.
.	.	.

Table 3.(c)

Table 3.(a),(b),(c) show the formation of Stego-Image.

The S\_Img1 and S\_Img2 pixel decimal grey values are stored by odd-even position respectively ( as table 3.(c) ) and remaining unaltered pixel grey values ( from pixel co-ordinate (0,8) as table 3.(c) ) of Stego-Key-Image are simply copied to form Stego-Image.

Sender transmits this Stego-Image to the receiver. The receiver uses the detector algorithm to extract the secret message from the stego image.

In the detector process, at the receiver side, the Stego-Image is shared in two parts with respect to odd-even position of its pixel values as table 4 .

Pixel Co-ordinate of Stego-Image	Stego-Image-pixel grey values	Equivalent 8-bit binary
(0,0)	163	10100011
(0,1)	161	10100011
(0,2)	158	10011110
(0,3)	157	10011111
(0,4)	156	10011100
(0,5)	153	10011101
(0,6)	155	10011011
(0,7)	161	10011011
(0,8)	168	10101000
(0,9)	173	10101101
.	.	.
.	.	.
.	.	.

Table 4.(a)

Pixel Co-ordinate of S_Img1	Pixel Co-ordinate of Stego-Image	S_Img1-pixel grey values	Equivalent 8-bit binary (B_SImg1)
(0,0)	(0,0)	163	10100011
(0,1)	(0,2)	158	10011110
(0,2)	(0,4)	156	10011100
(0,3)	(0,6)	155	10011011
(0,4)	(0,8)	168	10101000
.	.	.	.
.	.	.	.
.	.	.	.

Table 4.(b)

Pixel Co-ordinate of S_Img2	Pixel Co-ordinate of Stego-Image	S_Img2-pixel grey values	Equivalent 8-bit binary (B_SImg2)
(0,0)	(0,1)	161	10100011
(0,1)	(0,3)	157	10011111
(0,2)	(0,5)	153	10011101
(0,3)	(0,7)	161	10011011
(0,4)	(0,9)	173	10101101
.	.	.	.
.	.	.	.
.	.	.	.

Table 4.(c)

Table 4.(a),(b),(c) show the Stego\_Image sharing.

B\_SImg1 and B\_SImg2 are the equivalent eight-bit binary of S\_Img1 and S\_Img2 pixel grey values respectively.

The LSBs of B\_SKey1 and B\_SKey2 are bitwise-XORed with LSBs of B\_SImg1 and B\_SImg2 respectively as table 5.

LSB of B_SKey1 (l1)	S_Img1 (si1)	m1 = l1 ^ si1
0	1	1
0	0	0
0	0	0
0	1	1

Table 5.(a)

LSB of B_SKey2 (l2)	S_Img2 (si2)	m2 = l2 ^ si2
1	1	0
0	1	1
1	1	0
1	1	0

Table 5.(b)

Table 5.(a) shows the bitwise XOR operation between LSB of B\_SKey1 and S\_Img1 ; Table 5.(b) shows the bitwise XOR operation between LSB of B\_SKey2 and S\_Img2 .

The m1 and m2 are basically S\_Msg1 and S\_Msg2 respectively. The S\_Msg1 bit values are placed in odd position and S\_Msg2 bit values are placed in even position to form a sequence of zeros and ones as fig. 3.

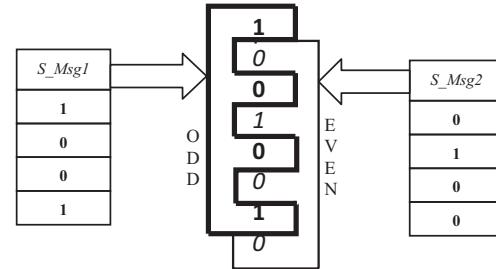


fig. 3 : Secret-Message formation

This sequence is sliced in eight bit block; each block is treated as an eight bit binary number. This binary number is stored in reverse way as ‘01001001’ and converted into its equivalent decimal.  $(01001001)_2 = (73)_{10}$ . This decimal number is treated as ASCII value, whose equivalent character is ‘I’. And secret textual message was “I”. So, the Secret\_Message is detected with the help of this prescribed method.

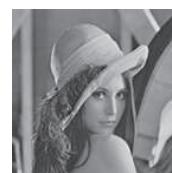


fig. 4.(a1)

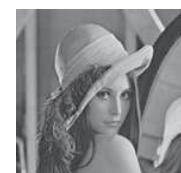


fig. 4.(a2)



fig. 4.(b1)



fig. 4.(b2)



fig. 4.(c1)



fig. 4.(c2)

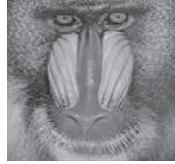


fig. 4.(d1)



fig. 4.(d2)



fig. 4.(e)

fig. 4.(a1),(b1),(c1),(d1) are the Stego\_Key image.  
 fig.4.(a2),(b2),(c2),(d2) are the Stego\_Image with embedded Secret\_Message “I” , “Computer” , “Department of Computer Science and Engineering - University of Kalyani” and the logo of fig.4.(e). All the Stego\_Key images and Stego\_Images are of dimension 100X100.

The Stego\_Key image and Stego\_Image can be compared using histogram analysis.

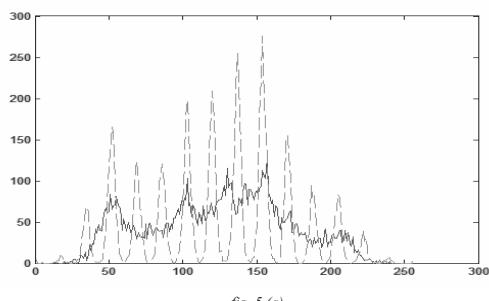


fig. 5.(a)

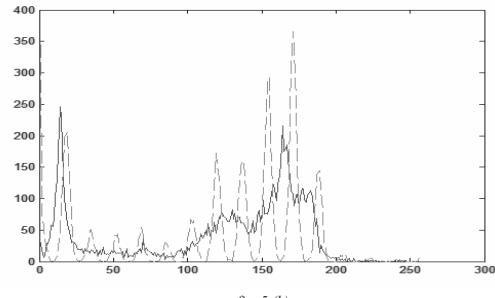


fig. 5.(b)

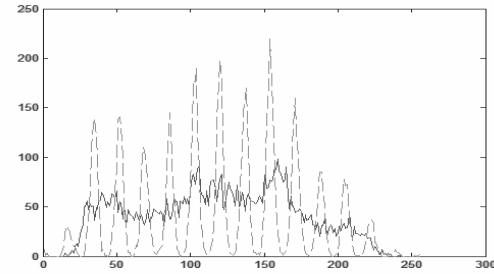


fig. 5.(c)

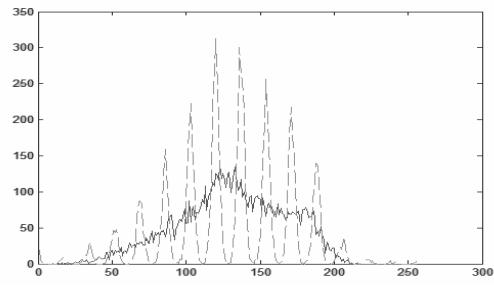


fig. 5.(d)

fig. 5.(a),(b),(c),(d) show the histogram comparison between Stego\_Key image (fig. 4.(a1),(b1),(c1),(d1) – shown by continuous line ) and Stego\_Image ( fig. 4.(a2),(b2),(c2),(d2) – shown by discontinuous line ).

From this histogram analysis of the Stego\_Key image and Stego\_Image, it is cleared that the secret messages are well embedded in the Stego\_Key image.

#### IV. CONCLUSION AND FUTURE WORK

The proposed methodology has been tested for huge amount of different textual and image secret messages, and it works satisfactory; it has high robustness, advanced level security and powerful embedding system. The secret message is hiding behind the stego-key-image, so the existence of the message is more secure. The bitwise XOR operation is done with only the LSBs of binary grey values of the stego-key-image, so the stego image pixel either remains same or increases (or, decreases) by one. The stego-key image is shared in two parts, so without getting proper image-shares the secret message cannot be detected. The next venture of this prescribed methodology is to make this genetic algorithm based steganographic system [19-20].

#### ACKNOWLEDGMENT

I would like to heartily thank Prof. Bidyut B. Chaudhuri, Head, Computer Vision and Pattern Recognition Unit, Indian

Statistical Institute, Kolkata 700108, India, for discussion various aspects of this paper.

#### REFERENCES

- [1] Wayner, Peter (2009). *Disappearing cryptography* 3rd Edition: information hiding: Steganography & Watermarking. Amsterdam: MK/Morgan Kaufmann Publishers. ISBN 978-0-12-374479-1.
- [2] Jim Reeds, *The Ciphers in Book III of Trithemius's Steganographia*, AT & T Labs-Research, New Jersey 07932 ,26 March 1998.
- [3] Petitcolas FAP, Anderson RJ, Kuhn MG, *Information Hiding: A survey*, *Proceedings of the IEEE (special issue)* 87 (7): 1062–78.doi:10.1109/5.771065. Retrieved 2008-09-02.
- [4] *The origin of Modern Steganography* , web link : <http://www.mikebarney.net/stego.html> .
- [5] Farid, H. *Detecting Steganographic Message in Digital Images*. Technical Report TR2001-412, Computer Science. Hnaover, NH: Dartmouth College,2001.
- [6] A.Joseph Raphael , Dr.V Sundaram, *Cryptography and Steganography – A Survey* , Int. J. Comp. Tech. Appl., Vol 2 (3) , 626-630 626 , ISSN : 2229-6093.
- [7] Westfeld, A. , and A. Pfitzmann. *Attacks on Steganographic Systems Breaking the Steganographic Utilities EzStego, Jsteg, Steganos, and S-Tools and Some Lessons Learned*. In Proc. Int. Workshop Information Hiding, Dresden, Germany, 1999, 61.
- [8] Fridrich, J. , M. Goljan, and D. Hoga. *Steganalysis of JPEG Images : Breaking the F5 Algorithm*. In Proc. Int. Workshop Information Hiding, Noordwijkerhout, Netherlands, 2002, 310.
- [9] Frank Y. Shih , *Digital Watermarking and Steganography : Fundamentals and Techniques*. CRC Press, Taylor & Francis Group, ISBN-13: 978-1-4200-4757-8.
- [10] Johnson, N., Z. Duric, and S. Jajodia, *Information Hiding : Steganography and Watermarking*, Boston : Kluwer Academic,2001.
- [11] Johnson, N., and S. Jajodia , *Exploring Steganography: Seeing the Unseen*, IEEE Computer 31 (1998): 26.
- [12] S. Hetzl, P. Mutzel, *A Graph Theoretic Approach to Steganography* , 9<sup>th</sup> IFIP , ICCMS, vol.3677, Salzburg, Australia, pp:119-128, 2005.
- [13] N.Provos, *Defending Against Statistical Steganalysis*, 10<sup>th</sup> USENIX security symposium, Washington DC,2001.
- [14] P.Sallee, *Model based Steganography*, IWDW 2003, LNCS Vol. 2939, pp:154-167,2003 .
- [15] Jessica Fridrich et.al, *Writing on Wet Paper*, ACM workshop on multimedia and security, Magdeburg, Germany, 2004.
- [16] Kaushal Solanki et.al, *Statistical Restoration for Robust and Secure Steganography*, ICIP, Vol.2, pp:1118-1121,2005.
- [17] Fridrich, J., M.Goljan and R.Du , *Reliable Detection of LSB Steganography in Color and Grayscale Images*. In Proc. ACM Workshop Multimedia Security, Ottawa, Ontario,2001.
- [18] Naor M., and Shamir A., *Advances in Cryptology -Eurocrypt '94* (A. DeSantis, Ed.), Lecture Notes in Computer Science, Vol. 950, pp. 1-12, Springer-Verlag, Berlin.1995.
- [19] Shih, F.Y. and Y.T. Wu. *Enhancement of Image Watermark Retrieval Based on Genetic Algorithm*. J. Visual Communication and Image Representation, 16(2005) : 115.
- [20] Shih, F.Y. and Y.T. Wu. *Combinatorial Image Watermarking in the Spatial and Frequency Domains*. Pattern Recognition 36(2003) : 969.