A Relevant Technique of Hiding Information in an image using Shared Cryptography

Boina Avinash Kumar
Scholar M.Tech
Department of CS&IT
Institute of Technical Education & Research, S'O'A University, Bhubaneswar, Odisha.

Madhusmita Das
Assistant Professor
Department of CS&IT
Institute of Technical Education & Research, S'O'A University, Bhubaneswar, Odisha.

Madhusmita Sahu
Assistant Professor
Department of CS&IT
Institute of Technical Education & Research, S'O'A University, Bhubaneswar, Odisha.

ABSTRACT

Steganography is the art and science of hidden writing. The purpose of steganography is covert communication to hide the existence of a message from an intermediate attacker .This paper will discuss about pure steganographic technique, which does not require prior exchange of data like shared-keys. The proposed technique use a secure mask generation technique and the masking of original data repeatedly generate shares. These shares of secret data are embedded using an irreversible stegonographic technique in cover image called stego image, which is sent to receiver node.

General Terms

Stegonography, Cryptography, Network Security

Keywords

Shared Cryptography, Masking Stegonography, Image Stegonography, Digital Steganography.

1. INTRODUCTION

The amount of digital images has increased rapidly on the internet. Image security becomes increasingly important for many applications, for example, confidential transmission, and video surveillance, military and medical applications. Steganography: literally "hidden writing". Now a days steganography is most often associated with embedding data in some form of electronic media[7],[8]. In steganography , the secret message is embedded in to an image (or any media) called cover image, and then sent to the reciever who extracts the secret message from the cover message. After embedding the secret message, the cover image is called a stego – image. This image should not be distinguishable from the cover image, so that the attacker cannot discover any embedded message. The difference between Steganography and the more commonly used cryptography is that while cryptography scrambles and obfuscates data that can then be accessed publicly(without consequence), Steganography conceals the data altogether [10],[15]. Data from a "covert", or source file is hidden by altering insignificant bits of information in an "covert", or host file[1],[4].

The security of hidden data can be obtained by two ways : Secret Share generation and steganography. A combination of two techniques can be used to increase the data security. The Secret Share generation is a technique used to scramble and obfuscates the secret data to be transmitted, by masking of original data repeatedly generate shares [3]. By using steganography the secret shares are concealed in the image and assuring that the visibility of image should not be distinguishable from the cover image, so that it almost becomes impossible for the attacker to discover any embedded messages. There are many techniques for encrypting data, which vary in their security, robustness, performance and so on. Also, there are many ways for embedding a message into another media. The most popular one is embedding a message into a coloured image using LSB. In this method the data is being hidden in the least significant bit of each pixel in the cover image [9],[12]. However, there are other known methods for hiding messages. However, there are many techniques for hiding a message in a binary image.

The recent interest in digital steganography is fueled by the increased amount of communication which is mediated by computers and by the numerous potential commercial applications. The hidden information could potentially be used to detect or limit the unauthorized propagation of the innocent looking / carrier data. Because of this, there have been numerous proposals for protocols to hide data in channels containing pictures, video, audio, and even typeset text. Many of these protocols are extremely clever and rely heavily on domain-specific properties of these channels.

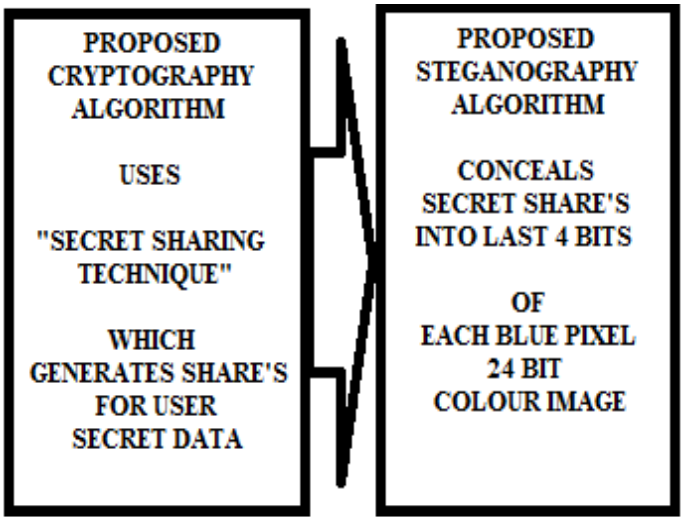
This paper presents a new scheme for embedding bits. This algorithm is used to hide secret share data in the last 4 bits of a blue pixel in 24 bit colour image that need to be modified to hide the data successfully. The proposed Algorithm emphasized on following characteristics are Design Simplicity, Enhanced Security, Resistance against all known attacks. Our goal is to propose a steganography mechanism that allows the hiding of a large quantity of data as possible in a coloured image without resulting much difference between the original image and the embedded image significantly. A combination of implementation, analysis and evaluation has to be done for hiding information in a colour image.

Section 2 consists of our scheme describes about proposed security technique algorithms. Section 3 describes about implementation of our scheme with examples. Finally conclusion is drawn in section 4,5 and Section 6 describes about future scope.

2. Proposed Security Technique

The proposed Security Technique of our scheme is divided into two Algorithms. The first algorithm is "Proposed Cryptographic Algorithm" and the second algorithm is "Proposed Steganographic Algorithm".

Fig.1 Proposed Model used in our scheme



2.1 Proposed Cryptography Algorithm

The proposed cryptographic Algorithm is used to generate shares of secrete data and Reconstruction of secrete data from those shares. This cryptographic algorithm consist of 4 steps.

Step 1. Mask generation

Step 2. Conversion of secrete data to binary form

Step 3. Share generation technique

Step 4. Secrete reconstruction from shares

The Secret Sharing Technique comprises of four basic steps for "generating Share's" and for "Secret Reconstruction from obtained shares". Before we generate share's, firstly we have to design the required masks and then convert the user secret data (i.e. text, audio, video) to the binary form, now perform AND operation repeatedly with the user secret data and masks to generate share's.

Step 1. Mask generation technique using n=5,k=3 values are fixed constants is as follows.

Step 1.1: List all row vectors of size **n** having the combination of **(k-1)** nos. of **0's** and **(n-k+1)** nos. of **1's** and arranges them in the form of a matrix. Now dimension of the matrix will be (**(nCk-1)-2) × n.**

Step 1.2: Transpose the matrix generated in Step-1.Now, each row of the transposed matrix will be the individual mask and the size of each mask is **((nCk-1)-2).** Now Dimension of the transposed matrix will be **n × ((nCk-1)-2)**.

Step 2. Converting the user secret data into binary form.

Step 2.1: Convert the secret data to the ASCII decimal value.

Step 2.2: Now convert the ASCII decimal value of secret data to binary value.

Step 3. Share generation technique.

On obtaining 'n' number of masks from "step-2 of mask generation technique",On performing AND operation repeatedly on every individual secret data with 'n' number of masks, generates ' n ' unique share's.

(eg : let secret data is a text message – 'Sai', there will be 3 * 5 number of shares, i.e 3 represents 'number of characters' and 5 represents 'number of masks'). (shares generated using masking of secrete data are embedded in an image using proposed stegonagraphic technique and the stego image is send to the receiver)

Step 4. Secret Reconstruction from Shares.

At receiver side, On obtaining at least k number of shares to every character in secret data , By performing OR operation on the received set of shares will generate original secret data. (eg : let secret data is a text message – 'Sai', on receiving 3 sets , each set containing 3 shares and on performing OR operation on each set gives the original secret data).

2.2 Proposed Steganography Algorithm

The proposed algorithm for steganography comprises of two basic Algorithms (**A).** Proposed Embedding Algorithm and (**B).** Proposed Extraction Algorithm

(A.) The **Proposed Embedding Algorithm** is comprises of three sub-algorithms 1. Algorithm for Preparing Cover or container image, 2. Algorithm for Preparing Secret text Message, 3. Algorithm for Preparing Stego Image.

1. Algorithm for Preparing Cover or container image.

Step 1: Read the 24 bit colour image, which is used as cover/container image to hide the secret Message.

Step 2: From the obtained cover image,read RGB colour of each pixel.(i.e Red values + Green values + Blue values) .

Step 3: Read the blue colour value of each pixel
(i.e which is in ASCII decimal value).

Step 4: Convert the blue colour value to stream of binary bits.

Step 5: Now by embedding zeros into the last 4 bits of blue colour value of each pixel.

Step 6: Repeat step-**2** to step-**5** until the "size of data × 10" (i.e10 represents, on masking with one character of secret data generates five shares and is stored in 5 × 2 no's of pixels).

Step 7: Two adjacent pixels are used to store one shar**e** of secrete data**.**

2. Algorithm for Preparing Secret text Message.

Step 1: From the proposed cryptographic technique, Shares of the secret data is obtain

Step 2: Read each share in 8 bit binary value.

Step 3: Take the 4 least significant bits from each share, i.e.by performing AND operation with $(15)_{10}$. **(**e.g. "(share) $_2$ AND $(0\ 0\ 0\ 0\ 1\ 1\ 1\ 1)_2$"**).**

Step 4: Take the 4 upper significant bits from each share by performing right shift operation with $(4)_{10}$.

3. Algorithm for Preparing Stego Image.

Step 1: Read the RGB colour of each pixel from the cover image.

Step 2: Now obtain the secret message prepared by algorithm 2 (Algorithm for preparing secrete text message) and add the secret to the cover image by applying OR operation. (i.e. embed the data bit to the last 4 bits of the colour blue of the pixel.)

Step 3: Write the above secret embedded pixels to the stego image.

Step 4: Repeat step 2 and step 3 until all the shares of secrete data bits are embedded in the pixel of stego image.

Step 5: After completly embedding the secret data to pixels of stego image, Write the rest pixels to stego image file as it is in cover image.

(B.) The **Proposed Extraction Algorithm** is used to extract the original secret data from the obtained stego image.

Step 1: Open stego image file in read mode.

Step 2: Read each pixel of the stego image.

Step 3: Take two adjacent pixels from the stego image, to reconstruct a single share of original secrete data.

Step 4: From the first pixel, on 8 bit of colour blue, perform left shift by 4 bits.

Step 5: From the second pixel on 8 bit of colour blue, perform AND operation with 15 to the second pixel's 8 bit of colour blue. i.e (8 bit of colour Blue)$_2$ AND $(00001111)_2$.

Step 6: Add the result of step 4 and step 5 together by OR operation and we get the binary value of the one share.

Step 7: Repeat step 3 to step 6, until "Size of the secret data * 5".

Step 8: After obtaining all the shares from stego image, perform 'Secret Reconstruction share technique of proposed cryptographic algorithm. To reconstruct the original secret data at least 3 shares are required at receiver end.

Step 9: After obtaining all the original secret data from the shares which is in binary form, convert to ASCII Decimal value and then convert to character.

Now the original secret data which got embedded in the image is extracted at the receiver side using this Extraction Algorithm. The proposed security technique is best understandable by suitable examples for Proposed Secret sharing technique and for Proposed Steganography technique in the next section.

The sender want to send confidential message to receiver by using the proposed security technique algorithm that we discussed in this paper. Considering the confidential message be " **Sai**", this paper gives example for single character " **S** " , and the same process is continued for all the characters of the confidential message .

3. EXAMPLE FOR PROPOSED SECURITY TECHNIQUE.

Example for share generation of the secrete data using proposed cryptographic algorithm.

Step 1: Mask generation technique using n=5, k=3 .

Step 1.1: List all row vectors of size **n** having the combination of **(k-1)** nos. of **0's** and **(n-k+1)** nos. of **1's** and arrange them in the form of a matrix.

$$[M] = \begin{bmatrix} 0\,0\,1\,1\,1 \\ 0\,1\,0\,1\,1 \\ 0\,1\,1\,0\,1 \\ 0\,1\,1\,1\,0 \\ 1\,0\,0\,1\,1 \\ 1\,0\,1\,0\,1 \\ 1\,0\,1\,1\,0 \\ 1\,1\,0\,0\,1 \end{bmatrix}$$

Now dimension of the matrix will be **((nCk-1)-2) × n .**

Step1.2: Transpose the matrix generated in Step-1.Now, each row of the transposed matrix will be the individual mask and the size of each mask is **((nCk-1)-2).**

$$[M]^T = \begin{matrix} 0\,0\,0\,0\,1\,1\,1\,1 & \text{(Mask - M1)} \\ 0\,1\,1\,1\,0\,0\,0\,1 & \text{(Mask - M2)} \\ 1\,0\,1\,1\,0\,1\,1\,0 & \text{(Mask - M3)} \\ 1\,1\,0\,1\,1\,0\,1\,0 & \text{(Mask - M4)} \\ 1\,1\,1\,0\,1\,1\,0\,1 & \text{(Mask - M5)} \end{matrix}$$

Now Dimension of the transposed matrix is **n × ((nCk-1)-2)**.

Step2: Conversion of secret data into binary form.

Let the user want to send the secret data be - Sai, So according to step 2 of proposed cryptographic algorithm, the binary equivalent of the secret text message is obtained.

Character	ASCII Decimal Value	Binary Value
S	83	0 1 0 1 0 0 1 1
a	97	0 1 1 0 0 0 0 1

Character	ASCII Decimal Value	Binary Value

i 105 0 1 1 0 1 0 0 1

Step 3: Share generation technique.

The binary value of secret data 'S' is 01010011, by using share generation technique 5 shares are generated by repeatedly performing AND operation with all the generated Masks in step 1. The share generation process for character S is as,

Secret Data - 0 1 0 1 0 0 1 1
Mask – M1 - 0 0 0 0 1 1 1 1 (AND operation)

Share 1 - 0 0 0 0 0 0 1 1

Similarly Share 2,3,4,5 are obtained in the following page.

Secret Data - 0 1 0 1 0 0 1 1 Mask – M2 - 0 1 1 1 0 0 0 1 Share 2 - 0 1 0 1 0 0 0 1	Secret Data - 0 1 0 1 0 0 1 1 Mask – M3 - 1 0 1 1 0 1 1 0 Share 3 - 0 0 0 1 0 0 1 0
Secret Data - 0 1 0 1 0 0 1 1 Mask – M4 - 1 1 0 1 1 0 1 0 Share 4 - 0 1 0 1 0 0 1 0	Secret Data - 0 1 0 1 0 0 1 1 Mask – M5 - 1 1 1 0 1 1 0 1 Share 5 - 0 1 0 0 0 0 0 1

Now 5 shares are generated for the secrete data 's' by applying proposed cryptographic algorithm. These shares are input to the proposed steganographic algorithm.

Example for embedding the shares of secrete data using Proposed algorithm for Steganography.

This technique is used to embed shares of the secret text message in an image.

(A.) Proposed embedding algorithm.

(1.) First cover image is prepared - We take 10 pixels of cover image for embedding single text character as for a single text character we create 5 shares by using proposed cryptographic algorithm and to hide each share we need 2 adjacent pixel, so to hide 5 shares of a single character, we take 10 pixels. Similarly for n text characters of secret message, we require n * 10 pixels to embed the secret text in the image.

10 sample pixels before embedding zeros in the blue values of 24 bit colour cover image.

1 0 1 1 0 1 0 0	0 0 1 0 1 1 1 0	1 1 0 1 0 0 0 1
0 0 1 1 1 1 1 1	1 1 1 0 0 0 0 0	0 0 0 1 1 1 0 1
0 1 0 1 1 1 1 1	1 1 0 0 0 0 0 1	0 0 0 0 0 1 1 1
0 1 1 0 1 1 1 1	1 0 0 0 0 0 1 1	0 0 0 0 1 0 1 1
0 1 1 1 0 1 1 1	0 0 0 0 0 1 1 1	0 0 0 1 0 0 1 1
0 1 1 1 1 0 1 1	1 0 1 0 0 0 0 1	0 0 1 0 0 0 1 1
0 1 1 1 1 1 0 1	1 0 0 1 0 0 0 1	0 1 0 0 0 0 1 1
0 1 1 1 1 1 1 0	1 0 0 0 1 0 0 1	1 0 0 0 0 0 1 1
1 0 0 1 1 1 1 1	1 0 0 0 0 1 0 1	1 0 0 0 0 1 0 1
1 0 1 0 1 1 1 1	1 0 0 0 0 0 1 1	1 0 0 0 1 0 0 1

After embedding zero's in the last 4 bit of blue values of 24 bit colour cover image .

1 0 1 1 0 1 0 0	0 0 1 0 1 1 1 0	1 1 0 1 0 0 0 0
0 0 1 1 1 1 1 1	1 1 1 0 0 0 0 0	0 0 0 1 0 0 0 0
0 1 0 1 1 1 1 1	1 1 0 0 0 0 0 1	0 0 0 0 0 0 0 0
0 1 1 0 1 1 1 1	1 0 0 0 0 0 1 1	0 0 0 0 0 0 0 0
0 1 1 1 0 1 1 1	0 0 0 0 0 1 1 1	0 0 0 1 0 0 0 0
0 1 1 1 1 0 1 1	1 0 1 0 0 0 0 1	0 0 1 0 0 0 0 0
0 1 1 1 1 1 0 1	1 0 0 1 0 0 0 1	0 1 0 0 0 0 0 0
0 1 1 1 1 1 1 0	1 0 0 0 1 0 0 1	1 0 0 0 0 0 0 0
1 0 0 1 1 1 1 1	1 0 0 0 0 1 0 1	1 0 0 0 0 0 0 0
1 0 1 0 1 1 1 1	1 0 0 0 0 0 1 1	1 0 0 0 0 0 0 0

After preparing cover image by embedded zeros, now obtain the shares from "**share generation technique**" of proposed cryptographic algorithm. And prepare the secret text message as stated in algorithm for preparing secrete text message .

(2.) Preparing Secret text Message.

The obtained secret shares from share generation technique for the text character 'S' on repeatedly performing AND operation are:

Share 1 - 0 0 0 0 0 0 1 1
Share 2 - 0 1 0 1 0 0 0 1
Share 3 - 0 0 0 1 0 0 1 0
Share 4 - 0 1 0 1 0 0 1 0
Share 5 - 0 1 0 0 0 0 0 1

Converting each share of 8 bits to two 4bits binary value and embed these values in the zero embedded cover image. This process is achieved by two steps 3 & 4 in the Algorithm for Preparing Secret text Message .

Considering for **share 1 - 0 0 0 0 0 0 1 1** to two 4 bits binary values .

From Step 3 of Algorithm for Preparing Secret text Message, Take the 4 least significant bits from each share, i.e by performing AND operation with $(15)_{10}$. (LSB)

$(0\ 0\ 0\ 0\ \ 0\ 0\ 1\ 1)_2$ AND $(0\ 0\ 0\ 0\ 1\ 1\ 1\ 1)_2 = 0\ 0\ 0\ 0\ \ 0\ 0\ 1\ 1$

From step 4 of Algorithm for Preparing Secret text Message, Take the 4 upper significant bits from each share and perform right shift operation by 4. (MSB)

$(0\ 0\ 0\ 0\ 0\ 0\ 1\ 1)_2 = 0\ 0\ 0\ 0\ \ 0\ 0\ 0\ 0$

Similarly perform the same process for rest of the shares as,

For Share 2 - 0 1 0 1 0 0 0 1 ,
$(0\ 1\ 0\ 1\ 0\ 0\ 0\ 1)_2$ AND $(0\ 0\ 0\ 0\ 1\ 1\ 1\ 1)_2 = 0\ 0\ 0\ 0\ \ 0\ 0\ 0\ 1$
and $(0\ 1\ 0\ 1\ 0\ 0\ 0\ 1)_2$ Shift to Right by 4 = $0\ 0\ 0\ 0\ \ 0\ 1\ 0\ 1$

For Share 3 - 0 0 0 1 0 0 1 0,

$(0\ 0\ 0\ 1\ 0\ 0\ 1\ 0)_2$ AND $(0\ 0\ 0\ 0\ 1\ 1\ 1\ 1)_2 = 0\ 0\ 0\ 0\ \ 0\ 0\ 1\ 0$
and $(0\ 0\ 0\ 1\ 0\ 0\ 1\ 0)_2$ Shift to Right by 4 = $0\ 0\ 0\ 0\ \ 0\ 0\ 0\ 1$

For Share 4 - 0 1 0 1 0 0 1 0 ,

(0 1 0 1 0 0 1 0)2 AND (0 0 0 0 1 1 1 1)2 = 0 0 0 0 0 0 1 0
and (0 1 0 1 0 0 1 0)2 Shift to Right by 4 = 0 0 0 0 0 1 0 1

For Share 5 - 0 1 0 0 0 0 0 1,

(0 1 0 0 0 0 0 1)2 AND (0 0 0 0 1 1 1 1)2 = 0 0 0 0 0 0 0 1
and (0 1 0 0 0 0 0 1)2 Shift to Right by 4 = 0 0 0 0 0 1 0 0

(3.) Preparing Stego image : This is prepared by performing OR operation on zero embedded cover image and prepared secret shares .The process for single character S is performed in fig 2.

This process is used to prepare rest of the characters in the secret text message. In our secret text message we have 3 characters, so we require totally 30 pixels to embed the 15. secret shares generated during Secret sharing technique. As each share is converted to two 4 bit values in order to accommodate perfectly in to the last 4 bits of blue value of 24 bit colour cover image. So twice the number of secret shares required for performing embedding process of secret in cover image to form stego image.

Fig 2: Preparing Stego Image by performing OR operation on zero embedded cover image & prepared secret shares A of 2.2 .

1 0 1 1 0 1 0 0	0 0 1 0 1 1 1 0	1 1 0 1 0 0 0 0	OR	0 0 0 0	0 0 0 0	=	1 0 1 1 0 1 0 0	0 0 1 0 1 1 1 0	1 1 0 1 0 0 0 0	Share 1
0 0 1 1 1 1 1 1	1 1 1 0 0 0 0 0	0 0 0 1 0 0 0 0	OR	0 0 0 0	0 0 1 1	=	0 0 1 1 1 1 1 1	1 1 1 0 0 0 0 0	0 0 0 1 0 0 1 1	
0 1 0 1 1 1 1 1	1 1 0 0 0 0 0 1	0 0 0 0 0 0 0 0	OR	0 0 0 0	0 1 0 1	=	0 1 0 1 1 1 1 1	1 1 0 0 0 0 0 1	0 0 0 0 0 1 0 1	Share 2
0 1 1 0 1 1 1 1	1 0 0 0 0 0 1 1	0 0 0 0 0 0 0 0	OR	0 0 0 0	0 0 0 1	=	0 1 1 0 1 1 1 1	1 0 0 0 0 0 1 1	0 0 0 0 0 0 0 1	
0 1 1 1 0 1 1 1	0 0 0 0 0 1 1 1	0 0 0 1 0 0 0 0	OR	0 0 0 0	0 0 0 1	=	0 1 1 1 0 1 1 1	0 0 0 0 0 1 1 1	0 0 0 1 0 0 0 1	Share 3
0 1 1 1 1 0 1 1	1 0 1 0 0 0 0 1	0 0 1 0 0 0 0 0	OR	0 0 0 0	0 0 1 0	=	0 1 1 1 1 0 1 1	1 0 1 0 0 0 0 1	0 0 1 0 0 0 1 0	
0 1 1 1 1 1 0 1	1 0 0 1 0 0 0 1	0 1 0 0 0 0 0 0	OR	0 0 0 0	0 1 0 1	=	0 1 1 1 1 1 0 1	1 0 0 1 0 0 0 1	0 1 0 0 0 1 0 1	Share 4
0 1 1 1 1 1 1 0	1 0 0 0 1 0 0 1	1 0 0 0 0 0 0 0	OR	0 0 0 0	0 0 1 0	=	0 1 1 1 1 1 1 0	1 0 0 0 1 0 0 1	1 0 0 0 0 0 1 0	
1 0 0 1 1 1 1 1	1 0 0 0 0 1 0 1	1 0 0 0 0 0 0 0	OR	0 0 0 0	0 1 0 0	=	1 0 0 1 1 1 1 1	1 0 0 0 0 1 0 1	1 0 0 0 0 1 0 0	Share 5
1 0 1 0 1 1 1 1	1 0 0 0 0 0 1 1	1 0 0 0 0 0 0 0	OR	0 0 0 0	0 0 0 1	=	1 0 1 0 1 1 1 1	1 0 0 0 0 0 1 1	1 0 0 0 0 0 0 1	

(B.) From the proposed extraction algorithm we perform steganaanalysis and extract the secret message sent by the sender.

On receiving the stego image from the sender, the receiver reads two adjacent pixels (Only last 8bits of two pixels) to retrieve a single individual shares of the secret text message which is in the form of shares that are embedded in the cover image.Similarly extraction of all other share is done. Extraction of shares is the reverse process of embedding secret shares in the cover, this process involves two vital steps to extract.

Firstly from the first pixel on 8 bit of colour blue, perform left shift by 4 bits. Secondly From the second pixel on 8 bit of colour blue, perform AND operation with 15 to the second pixel's 8 bit of colour blue. i.e (8 bit of colour Blue)2 AND (00001111)2 ."
Finally the result obtained from above two steps together by OR operation and we get the binary value of the each share. Repeat these steps, until all the shares are reconstructed.

Generation of Share 1:

1.) 1 1 0 1 0 0 0 0 left shift by 4 bits = **0 0 0 0 0 0 0 0**

2.) 0 0 0 1 0 0 1 1 AND with15 (00001111)2 = **0 0 0 0 0 0 1 1**

3.) (0 0 0 0 0 0 0 0) OR (0 0 0 0 0 0 1 1) = 0 0 0 0 0 0 1 1

Generation of Share 2:

1.) 0 0 0 0 0 1 0 1 left shift by 4 bits = **0 1 0 1 0 0 0 0**

2.) 0 0 0 0 0 0 0 1 AND with15 (00001111)2 = **0 0 0 0 0 0 0 1**

3.) (0 1 0 1 0 0 0 0) OR (0 0 0 0 0 0 0 1) = 0 1 0 1 0 0 0 1

Generation of Share 3:

1.) 0 0 0 1 0 0 0 1 left shift by 4 bits = **0 0 0 1 0 0 0 0**

2.) 0 0 1 0 0 0 1 0 AND with15 (00001111)2 = **0 0 0 0 0 0 1 0**

3.) (0 0 0 1 0 0 0 0) OR (0 0 0 0 0 0 1 0) = 0 0 0 1 0 0 1 0

Generation of Share 4:

1.) 0 1 0 0 0 1 0 1 left shift by 4 bits = **0 1 0 1 0 0 0 0**

2.) 1 0 0 0 0 0 1 0 AND with15 (00001111)2 = **0 0 0 0 0 0 1 0**

3.) (0 1 0 1 0 0 0 0) OR (0 0 0 0 0 0 1 0) = 0 1 0 1 0 0 1 0

Generation of Share 5:

1.) 1 0 0 1 0 1 0 0 left shift by 4 bits = **0 1 0 0 0 0 0 0**

2.) 1 0 0 0 0 0 0 1 AND with15 (00001111)2 = **0 0 0 0 0 0 0 1**

3.) (0 1 0 0 0 0 0 0) OR (0 0 0 0 0 0 0 1) = 0 1 0 0 0 0 0 1

With this we obtained all the shares of the secrete message by using "Proposed Extraction Algorithm" from stego image.,

'Secret Reconstruction technique' is applied at the receiver end to reconstruct the original message from the shares. At the receiver end at least 3 share must be received to reconstruct the original message.

Secret Reconstruction for the set of shares 1, 2, 3 & set of shares 3, 4, 5 are :

Share 1 - **0 0 0 0 0 0 1 1**	Share 3 - **0 0 0 1 0 0 1 0**
Share 2 - **0 1 0 1 0 0 0 1**	Share 4 - **0 1 0 1 0 0 1 0**
Share 3 - **0 0 0 1 0 0 1 0**	Share 5 - **0 1 0 0 0 0 0 1**
Secret - 0 1 0 1 0 0 1 1 (83 – S)	Secret- 0 1 0 1 0 0 1 1

Secret Reconstruction for the set of shares 1,3,5 is :

Share 1 - **0 0 0 0 0 0 1 1**

Share 3 - **0 0 0 1 0 0 1 0**

Share 5 - **0 1 0 0 0 0 0 1**

Secret - 0 1 0 1 0 0 1 1 (83 – 'S' Reconstructed)

Similarly we can obtain secret text message Sai, by using the above process. Secret reconstruction for text characters "a"& "i" are as follows.

Share 1 – **0 0 0 0 0 0 0 1**	Share 1 – **0 0 0 0 1 0 0 1**
Share 2 - **0 1 1 0 0 0 0 1**	Share 2 - **0 1 1 0 0 0 0 0**
Share 3 - **0 0 1 0 0 0 0 0**	Share 3 - **0 0 1 0 0 0 0 0**
Secret - 0 1 1 0 0 0 0 1 (97)	Secret - 0 1 1 0 1 0 0 1 (105)

In this we reconstructed 'a' & 'i' whose ASCII values are 97 & 105, the same secret sharing cryptographic technique and steganographic technique is used for embedding as well as for extracting & reconstructing secret message "Sai" from cover image.

This Proposed Security Technique was implemented by using MAT LAB (R2010a), taking input as text message and on processing through the proposed cryptographic algorithm phase and proposed stegonagraphic algorithm phase generates a stego image or encoded image as an output.

By taking input as 'Sai', a text word which is embedded in the selected image known as cover image or original image and on processing through the proposed security technique generates an encoded image.

Secret Text data as Input is **Sai** (3 characters to be hidden in the cover image).

Fig 3(a): Original Image (4face_cover.jpg)

Fig 3(b): Encoded Image (4face_stego.jpg)

Output after performing Proposed Security Technique is : Stego or Encoded Image (4face_stego.jpg).

Message extracted from the Stego image – Sai

Another example, taking input as 5,371 secret text characters to be hidden in the cover image and on processing through several steps of proposed security technique generates an encoded or stego image as output.

Fig 4(a): Original Image (dog_cover.jpg)

Fig 4(b): Encoded Image (dog_stego.jpg)

Output after performing Proposed Security Technique is : Stego or Encoded Image (dog_stego.jpg). The message extracted from the Stego image consists of 5,371 characters forms the original secret text data.

Finally by embedding maximum number of words i.e. 27,200 ASCII characters in to "saimom.jpg" cover image and on processing through the proposed security technique generates an encoded image known as stego image, shown in fig.5(a, b).

Fig. 5(a): Original Image – before embedding 27,200 words.

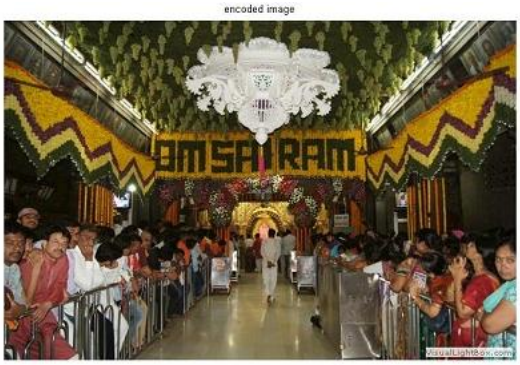


Fig 5(b): Stego Image after embedding 27,200 words in cover

4. EXPERIMENTAL RESULTS

Several experiments have been done to examine the performance of the proposed security technique which uses both (secret sharing) shared cryptography and steganography. Many standard color images with different textural properties were taken as the cover-images shown in Fig. 3, 4 & 5. Our experiments are described in two major parts: capacity test and security test (i.e. imperceptibility test).

Imperceptibility test: The imperceptibility is evaluated by the objective quality measurement PSNR (peak signal to noise ratio)

$$\text{PSNR} = 10 \times \log\left(\frac{255^2}{\text{MSE}}\right) \text{(dB)},$$

Where MSE represents the mean square error between the cover image x and the stego-image y:

$$\text{MSE} = \left(\frac{1}{512 \times 512}\right) \sum_{i=1}^{512} \sum_{j=1}^{512} (x_{ij} - y_{ij})^2.$$

Where: X_{ij} is the i^{th} row and the j^{th} column pixel in the original (cover) image,
Y_{ij} is the i^{th} row and the j^{th} column pixel in the encodded (stego) image,

M and N are the height and the width of the image,

I is the dynamic range of pixel values, or the maximum value that a pixel can take, for 8-bit images: I=255.

However, the MSE for colour images is defined as follows.

$$MSE_{AVG} = \frac{MSE_R + MSE_G + MSE_B}{3}$$

Where: MSE_R, MSE_G, and MSE_B are the MSE of red, green, and blue components respectively.

Thus, the best image quality can be found when the MSE value is very small or going to be zero since the difference between the original and encoded image is negligible. However, PSNR values between 20 and 40 can be considered as typical values [6,7]. Moreover, the higher the PSNR value of a stego image, the better the degree of hidden message imperceptibility**.**

Table.1 presents the PSNR values of the stego-images created by the proposed method using various embedding rates. In Table 1, the embedding rate is defined as

$$\text{Embedding Rate} = \frac{\text{Actual Embedding Bits.}}{\text{Embedding Capacity}}$$

It is clear that even the full capacities of the cover-images are used to embed the secret bits, the values of PSNRs are still higher than 30 dB It means that the proposed steganographic method can provide good imperceptibility performance.

Table1 1: The PSNR values (in dB) of the stego images created by the proposed method using various embedding rates.

Test Image	Embedding Rate				
	20%	**40%**	**60%**	**80%**	**100%**
4face.jpg	61.3654	61.3083	61.2504	61.1805	61.1213
dog.jpg	68.1475	67.9524	67.7559	67.5945	67.4153
saimom.jpg	75.0803	74.4880	74.0035	73.2297	72.9735

Table 2: The MSE (Mean Square Error) of the stego images created by the proposed method using various embedding rates.

Test Image	Embedding Rate				
	20%	**40%**	**60%**	**80%**	**100%**
4face.jpg	140.609	141.414	142.236	143.234	144.083
dog.jpg	71.362	72.768	74.212	75.420	76.783
saimom.jpg	35.676	37.853	39.732	42.929	44.043

Table 3: The number of characters (in words) embedded in stego images created by the proposed method using various embedding rates.

Test Image	Embedding Rate				
	20%	**40%**	**60%**	**80%**	**100%**
4face.jpg	806	1440	2160	2880	3600
dog.jpg	2030	4061	6091	8122	10152
saimom.jpg	5440	10923	16420	25939	27200

It is clear from the results that by increasing the number of characters the PSNR value decays. All the images are of same size, if we increase image size and number of characters remains constant, then the PSNR value will improve.

To compare with the other steganographic methods, we embedded the same amount of secret bits (embedding capacity of 20%) in various cover-images using the **"**4bit Simple LSB embedding technique**"** [4], **"**Hiding relevant information in an image**"** [3] and **"**the proposed method**"**.

("4bit Simple LSB" – which does not use any encryption on secret data, i.e. the secret data is directly embedded in the last 4 bits of the blue value of 24 bit pixel of color image**)** [4]. **(**"Hiding relevant information in an image**"** - which uses substitution and transposition cipher techniques for encrypting secret data and then embedded in the last 4bits of the blue value of 24 bit pixel of color image) [3].

Table 4 presents the comparison of PSNR values of the stego images created by various steganographic methods.
Table 4: Comparison of PSNR values (in dB) of the stego images created by various embedding algorithms.

Test Image	Embedding Algorithm		
	Proposed Method	[3] Hiding Relevant Information in an Image	[4] 4bit Simple LSB Embedding Technique
4face.jpg	61.4610	61.4488	61.4121
dog.jpg	68.3465	68.3846	68.3565
saimom.jpg	75.6224	50.5163	75.6171

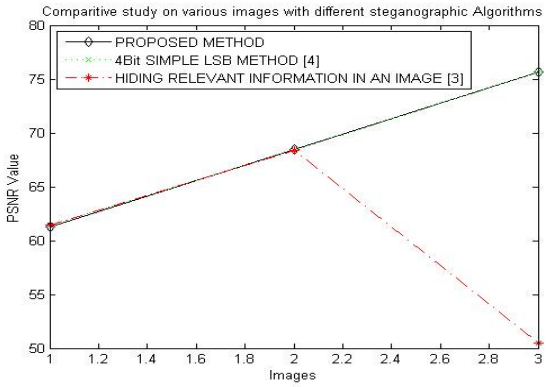


Fig. 6: Comparison of PSNR values of the stego images created by various embedding algorithms.

It is clear in the proposed method, the PSNR values of the stego-images created by the proposed method are only slightly lower than those created by the other steganographic methods.

5. CONCLUSION

This proposed Algorithm has following advantages:

i. Minimal change is permitted in Stego Image and normal human beings' eyes cannot catch any difference due to high rate of imperceptibility achieved.

ii. This new algorithm does not need any secret key.

iii. This proposed security technique is using both shared cryptographic technique and steganographic technique, provides high security to exchange of confidential text data over public channel in a safe way.

The 4-LSB substitutions of blue colour in a 24 bit pixel is a good method for embedding an acceptable amount of data. In the proposed algorithm embedded of data can easily be implemented and do not visually degrade the image to the point of being noticeable. Furthermore the encoded message can be easily recovered and even altered by a 3rd party.

6. FUTURE SCOPE

A modification to the proposed algorithm can increase further security to data hide. Statistical analysis of 0s and 1s can be applied block by block on data bit stream to make the algorithm more complex and we can achieve a new methodology to hide data in a more secured way.

Our future work is to distribute the information in the image randomly where a random generator can generate the location of the letter randomly.

7. REFERENCES

[1] F.A.P.Petitcalas, R.J.Anderson and M.G.Kuhn. "Information Hiding – A Survey". Proceedings of the IEEE Special issue on Protection of Multimedia Content. 87(7). 1062-1078. July 1999. DOI:10.1109/5.771065.

[2] D.Kahn. "The Code Breakers- The Story of Secret Writing". New York U.S.A: Scrinber. 1996.ISBN 0-684-83130-9.

[3] Madhusmita Das, Mahamaya Mohanty, "Hiding relevant information in an image" Information Processing and Management Communications in Computer and Information Science Volume 70, 2010, pp 501-504 .

[4] Alkhraisat Habes. "Information Hiding in BMP image implementation, Analysis and evaluation", Information Transmissions In Computer Networks - 2006,.

[5] Abhijit Das, Soumya Sankar Basu, Atal Chaudhuri, "A Novel Security Scheme for Wireless Adhoc Network", 978-1-4577-0787-2/11 IEEE 2011.

[6] Almohammad, A., and Ghinea, G., (2010). "Stego Image Quality and the Reliability of PSNR". International Conference on Image Processing Theory, Tools and Applications (IPTA 2010), Paris, France, 7-10 July,2010, 215-220, IEEE.

[7] Yuan-Hui Yu, Chin-Chen Chang, Iuon-Chang Lin, A new steganographic method for color and grayscale image hiding. Journal *Computer Vision and Image Understanding*, Volume 107, Issue 3, September 2007, Pages 183–194

[8] W.Bender, D.Gruhl and N.Morimoto, A.Lu. "Techniques for Data Hiding" Tech.Report. Mit Media Lab.1994.

[9] W.Bender, D.Gruhl, N.Morimoto, A.Lu. "Techniques for Data Hiding". IBM Systems Journal. Vol 35, issue 3-4, 1996. 313-336. ISSN: 0018-8670.

[10] R.H.Alwan, F.J.Kadhim, A.T.Al-Taani. "Data Embedding Based on Better Use of Bits in Image Pixels". International Journal of Signal Processing. Volume. 2, Number. 1, 2005. pp: 104-107. ISSN: 1304-4494.

[11] A.T.Al-Taani, A.M. Al-Issa. "A Novel Steganographic Method for Gray-Level Images". International Journal of computer, Information, and systems Science and Engeering.3:1.2009.pp:5-10 ISSN:2070-3732.

[12] M.Y.Wu ad J.H.Lee. "A Novel Data embedding Method for Two Color Facsimile Images". In proceedings of International Symposium on Multimedia Information Processing. Chung-Li Taiwan, R.O.C. 14-16 December 1998.

[13] Eric Cole. "Hiding in plain sight". Wiley Publishing, Indianapolis, Indiana, 2003.

[14] Y. Desmedt "Some recent research aspects of threshold cryptography" Proc of ISW.97 1st International Information Security Workshop vol.1196 of LNCS pp 158-173 Springer-Verlag 1997.

[15] A. Shamir: "How to share a secret?" Communications ACM, 22(11): pp612-613, 1979.