Multimed Tools Appl (2017) 76:24063–24090
DOI 10.1007/s11042-016-4111-x

A novel hybrid security mechanism for data communication networks

Neha Tayal[1] **· Ritesh Bansal**[1] **· Sangeeta Dhal**[1] **·**
Shailender Gupta[1]

Received: 24 May 2016 / Revised: 13 September 2016 / Accepted: 1 November 2016 /
Published online: 28 November 2016

Abstract With the growing internet technology over the last decade, the number of intruders trying to steal the confidential information has also risen. As a result, for the protection of secret data from unwanted access, hybrid security mechanisms employing the use of steganography to hide the encrypted data are gaining popularity. These mechanisms provide an extra level of security to the data. This paper proposes a hybrid mechanism that not only aims at providing good security but at the same time has high data embedding capacity and entropy values with low execution time complexity. To enhance the embedding capacity of the overall system, Improved Bit Plane Complex Steganography (IBPCS) along-with Huffman coding is used and for providing randomness, the use of chaos process wherever possible is done. The cryptographic technique employed is hierarchical visual cryptography due to its efficiency over other cryptographic mechanisms. The overall scheme is implemented in MATLAB-10 and the results prove that the proposed mechanism is efficient to other available schemes in literature.

Keywords Compression · Cryptography · Entropy · Steganography

✉ Neha Tayal
nehatayal2292@gmail.com

Ritesh Bansal
ritesh.bansal@hotmail.com

Sangeeta Dhal
sangeeta_dhall@yahoo.co.in

Shailender Gupta
shailender81@gmail.com

[1] YMCA University of Science and Technology, Faridabad 121002, India

1 Introduction

The need for security has arisen in the last few years with the growth in number of attacks by hackers to filch the confidential information [9]. Hence, researchers keep on searching and developing novel security mechanism to curb these attacks. Two most prominent solutions are cryptography [5] and steganography [14]. Cryptography can be defined as the encoding of secret data into a form which can only be read by the intended user. It can be categorized into two categories [26]: Symmetric key and Asymmetric key cryptography. In the former case, same key is used for encryption/decryption. The latter uses two keys: public and private key. The public key is known to all and is used for encryption purpose while private key is used for decrypting the message. Schemes based on former are considered to be computationally fast to those based on later that in contrast provide much better security.

The other available security mechanism in literature is steganography. It doesn't encode the data rather it hides the data into a cover media which can be an image, video etc. It can be broadly classified into two domains [22]: Spatial (Time) domain and Transform (Frequency) domain. The former techniques preserve picture quality and have high PSNR value but are not robust to attacks, while transform domain techniques provide robustness with the drawback of poor picture quality.

Using standalone technique i.e. steganography or cryptography is not sufficient, as the attacks have grown more sophisticated [26]. Therefore, researchers have tried to combine both with the aim to provide better security. This mechanism not only encodes the data but also hides the secret data as well (see Fig. 1), making it much more difficult for the intruder to retrieve the original secret data.

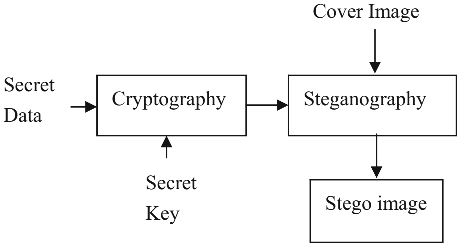
This paper proposes a new hybrid mechanism which tries to inculcate the advantages of both cryptography and steganography. The rest of the paper is organized as follows: Section 2 gives the detailed analysis of previously developed hybrid approaches and the objectives taken into consideration while section 3 discusses the proposed approach. The simulation set up parameters used to evaluate the efficacy of the proposed approach is discussed in section 4 followed by results in section 5. Overall comparison and conclusion is discussed in section 6.

2 Literature survey

Researchers have tried various permutation and combination of steganography and cryptography mechanism in order to achieve the following objectives as given below [1, 4, 7, 8, 11, 14–16, 18, 19, 21, 24, 27]:

- High PSNR
- Good Entropy value
- Low computational time

Fig. 1 Block Diagram of Hybrid Security Mechanism



- High data embedding capacity
- Good Security
- High Robustness

A brief description about various mechanisms [1, 4, 7, 8, 11, 14–16, 18, 19, 21, 24, 27] proposed in the recent past is given in Table 1.

After analyzing the details in Table 1, it is found that different hybrid mechanisms lead to different strengths and some overheads. Some are better in terms of security and some in terms of time complexity but there is no hybrid security mechanism which has very high embedding capacity. Though Divya Chaudhary et al. [1] has applied Huffman compression technique to increase size of embedding data still there is a need for a mechanism which can further improves embedding capacity while maintaining acceptable picture quality. Also, there is no focus on entropy in any mechanism which in turn increases the unpredictability of data. The literature shows that employing visual cryptography improves confidentiality [12] of data with marginal increase in time complexity as discussed by Divya Chaudhary et.al. [1]. This technique can further be improved by the use of hierarchical visual cryptography [17] instead of plain Visual cryptography. The next section gives the detailed information of the proposed technique keeping the above objectives into consideration.

3 The proposal

Basic architecture of the proposed technique for the sender's side is described in Fig. 2a. The secret message i.e. plain text is first compressed using Huffman *compression* then a *cryptographic cipher* is used to encrypt this compressed data. S*teganography* technique is then applied on this compressed encrypted data to hide its presence.

Figure 2b shows the process at the receiver side. On the stego-image *inverse steganography* technique is applied in order to extract the compressed encrypted data. Finally, the inverse decryption and de-compression mechanisms are applied in order to retrieve the plain textual data.

The detail explanation of each step is explained in the subsequent sub-sections.

3.1 Huffman compression

The purpose of using compression technique is to reduce the size of plain text as much as possible that in turn increases embedding capacity. Huffman compression is used in this paper due to its lossless nature and high compression ratio value [6, 20]. This technique assigns variable-length codes to data on the basis of its frequencies of occurrence in data. Larger the frequency, lower the encoding bits for a value. Hence most frequent data is compressed more. Before explaining Huffman compression in detail, few assumptions are made which are as follows:

Suppose the data to be compressed is arranged in array $arr=[A_1\ A_2\ A_3 .... A_n]$. Unique symbols used in this array are grouped in another array named as $symbol= [S_1\ S_2 ......S_n]$.*Z* is an array which is used to store the frequency of occurrence of each unique symbol. The probability of each unique element is calculated and stored in another array $p= [P_1\ P_2 ...P_n]$. *Huffmandict(), huffmandeco() and huffmanenco()* are inbuilt functions in MATLAB which are used for compression, *huffmandict()* is used to generate dictionary, which uses probability array (*p*) and unique symbol array (*symbol*) as input while *huffmanenco()* is used to compress the input data having data array (*arr*) and *huffmandict ()*output as input. The *huffmandeco()* is used to decompress the compressed data.

Table 1 Literature Survey

Author	Year	Encryption Scheme implemented	Steganography Scheme implemented	Advantages	Disadvantages
Piyush Marwah et al. [13]	2010	DES	Visual Cryptography	High PSNR High Correlation	Low Key Space, Low Security Limited Embedding Capacity
S.M. Masud Karim, et al. [14]	2011	Encryption using secret key	Modified LSB substitution Steganography	Higher PSNR value Good security	High Time Complexity Secret key has to be chosen properly
S Usha et al. [27]	2011	Play Fair Cipher and AES	LSB	High PSNR Good Security	Limited Embedding Capacity Low key space Complex Structure
Gokul M., et al. [4]	2012	Visual Cryptography	LSB substitution Steganography	Low Time complexity	Low PSNR Low Security
Shailender Gupta, et al. [21]	2012	RSA	LSB substitution Steganography	High Security Moderate PSNR High Key Space	Limited Embedding Capacity High Time Complexity
Mohammad A., et al. [15]	2012	Diffie Hellman	LSB	Moderate Security Moderate PSNR	Limited Embedding Capacity Moderate Time Complexity Low Key Space
R.Nivedhitha, et al. [16]	2012	DES	Least Significant Bit	Moderate PSNR	High Time complexity Limited Embedding Capacity Low security and Key Space
Ramakrishna Mathe, et al. [19]	2012	Diffie Hellman	LSB substitution Steganography	Moderate PSNR Better Security	Limited Embedding Capacity High Time Complexity
Lokesh Kumar [11]	2012	AES	Alteration Component Technique	Moderate Security High PSNR	Limited Embedding Capacity Low key space Complex Structure
Md. Rashedul Islam, et al. [7]	2014	AES	LSB substitution Steganography using Status bit	Good PSNR Good Security	High Time complexity Limited Embedding Capacity

Table 1 (continued)

Author	Year	Encryption Scheme implemented	Steganography Scheme implemented	Advantages	Disadvantages
PyePyeAung,et al. [18]	2014	AES	DCT	Enhanced Security Robustness	Distortion is high High time complexity Limited Embedding Capacity
ShingoteParshuramN., et al. [24]	2014	AES	LSB substitution Steganography	High PSNR value Good security	High Time Complexity Limited Embedding Capacity
DivyaChaudhary., et al. [1]	2015	Visual cryptography	Status LSB substitution Steganography	Good embedding capacity as Huffman Compression is used, High PSNR	High time complexity
SangeetaDhall.,et al. [8]	Accepted for publication	Visual cryptography	DCT	Good Security, Robustness	Limited embedding capacity

Algorithm for Huffman Compression technique (Sender Side)

```
Step 1: Data to be encoded is, arr=[A1 A2 A3…. An].

Step 2: Symbols used in the data, symbol=[S1 S2…..Sn].

Step 3: Take an empty array Z which represents
frequency of each data item and is initialized with all
zeroes.
          for i=1 to length(arr)
             Z(i)=0;
           end
Step 4: Calculate the frequency of each data item and
store it in array Z which is updated for every
occurrence.
          for i=1 to length(arr)
               k=arr(i);
               Z(k-64)=Z(k-64)+1;
           end
Step 5: Calculate the probability of each symbol present
in data by dividing number of occurrences by length of
data.
          p=Z/length(arr);
Step 6: Probability of each data signal is represented by
array p=[P1 P2…Pn].

Step 7: Create a dictionary using function huffmandict
which takes symbol and p arrays as arguments.

dict=huffmandict(symbol,p);

Step 8: Huffman encoded and compressed bits are
generated by using function huffmanenco with the help
of array    arr and dict.
compressed_data = huffmanenco (arr, dict).
 /* compressed as well as encoded bits are generated */
Step 9: Transmit compressed_data.
```

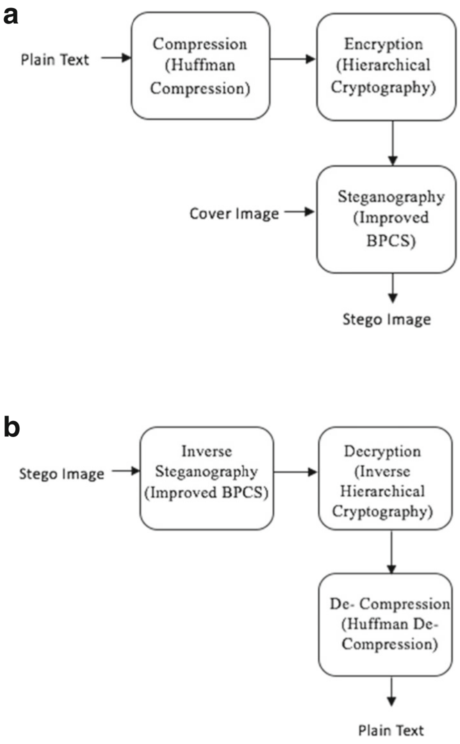

a
Plain Text
Compression (Huffman Compression)
Encryption (Hierarchical Cryptography)
Cover Image
Steganography (Improved BPCS)
Stego Image
b
Stego Image
Inverse Steganography (Improved BPCS)
Decryption (Inverse Hierarchical Cryptography)
De- Compression (Huffman De- Compression)
Plain Text

Fig. 2 **a** Block diagram of Proposed Hybrid Technique for sender side **b** Block diagram of Proposed Hybrid Technique for Receiver side

At the time of decoding, the same dictionary *dict* has to be generated and then the compressed and encoded data *compressed_data* and dictionary *dict* is passed as arguments in Huffman decoding function *huffmandeco* to extract the original message.

Algorithm for Huffman Compression technique (Receiver Side)

Step 1: Get the encoded message compressed_data.

Step 2: Generate same dictionary as generated at the sender side by using Huffman dict function which takes symbol and probability p arrays as its arguments.

dict=huffmandict(symbol, p);

Step 3: Uncompressed and decoded data bits are extracted by passing dict and encoded message compressed_data into Huffman decoding function huffmandeco.

Message = huffmandeco(compressed_data, dict).

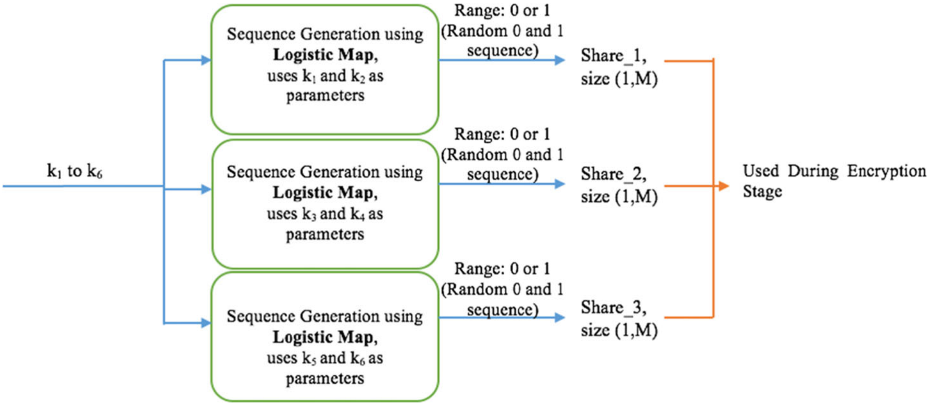

k₁ to k₆
Sequence Generation using Logistic Map, uses k₁ and k₂ as parameters
Sequence Generation using Logistic Map, uses k₃ and k₄ as parameters
Sequence Generation using Logistic Map, uses k₅ and k₆ as parameters
Range: 0 or 1 (Random 0 and 1 sequence)
Range: 0 or 1 (Random 0 and 1 sequence)
Range: 0 or 1 (Random 0 and 1 sequence)
Share_1, size (1,M)
Share_2, size (1,M)
Share_3, size (1,M)
Used During Encryption Stage

Fig. 3 Chaotic Key Generation

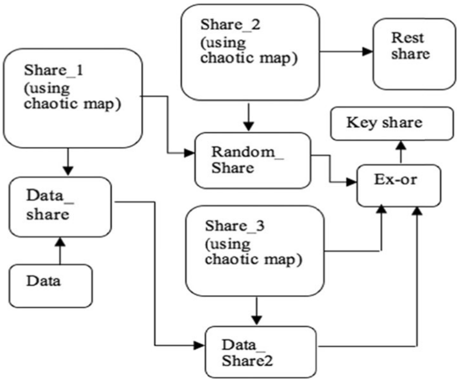
3.2 Cryptography technique used

After first step of compression, next step in the proposed mechanism is encryption process. In a three-layer security mechanism this is second layer. For the purpose of encryption, hierarchical visual cryptography technique is used [17, 28] because of it's simplicity with low time complexity and high security [2]. Before explaining encryption in detail, key generation step is explained, which is used during the process of encryption.

3.2.1 Key generation

For key generation, method used is chaos theory. A chaotic map is that which deals with non linear dynamical values where dynamical means the value changes over time

Fig. 4 Hierarchical cryptography encryption technique


Share_2 (using chaotic map)
Rest share
Share_1 (using chaotic map)
Key share
Random_ Share
Ex-or
Data_ share
Share_3 (using chaotic map)
Data
Data_ Share2

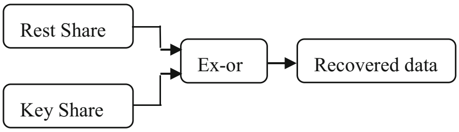


Fig. 5 Hierarchical cryptography decryption scheme

based on its current state. This map includes generation of random sequence for various 1D and 2D discrete maps based on mathematical equations and relations i.e. Logistic map, Cubic map, Ricker's map, Sin map, Henon map, Gingerbreadman map, Burgers' map, Tinkerbell map, etc.

One of the most studied examples of a one-dimensional system is logarithmic map, its properties and chaotic performance is also similar to logistic map [23]. Its equation is

$$x_{n+1} = A\ x_n\ (1 - x_n)$$

where x_n is a number between 0 and 1 and A lies in the interval (3.57,4). It is called Logistic map because it maps the population value at any time step to its value at the next time step. The properties like randomness and unpredictability enhances with the usage of logistic map. Chaotic maps are highly sensitive to initial condition as a slight change in initial condition changes the population values.

From Fig. 3, Three chaotic sequences, Share_1, Share_2 and Share_3 are generated using logistic chaotic equation and these sequences will be further used in encryption process. *M* is the length of the compressed data. K_1 to K_6 are the initial constants required to generate encryption keys, there values are given in set up parameters.

4 Encryption process

In this scheme, data is converted into 2 Shares – Rest Share and Key share (see Fig. 4), where a share is defined as a component of data which contains partial information and appears as a noise to unauthorized users. First Share_1 which is generated as part of key generation, is used to generate another sequence (Data_Share) based on the idea: if data bit is 1, compliment of Share_1 is written into Data_Share else bit of Share_1 is copied into Data_Share.

As the name suggests, hierarchical relationship is followed, hence three shares, Random_Share, Share_3, Data_Share2, are further generated on the basis of same functionality as stated above, and these three shares are xored to generate the Key Share. Share_2 will act as Rest Share. These two resulted shares are then transmitted further by the sender.

Algorithm for Hierarchical Visual Cryptography (Sender Side)

Step 1: Get the compressed message and convert it into binary form.

Step 2: Generate sequences of random numbers, Share_1, Share_2 and Share_3 having length same as message using chaotic function i.e. Logistic map.

Step 3: Generate Data_share by using Share_1 and message data. If message bit is 1, compliment of bit of Share_1 is written into Data_share else the bit of Share_1 is copied as it is in Data_share.

```
            for i=1 to length(message)
                 if(data(i)==0)
Data_share(i)=Share_1(i);
                 else
Data_share(i)=not(Share_1(i));
                end
            end
```

Step 4: Now, to generate Random_share and Data_share2, same methodology is employed,

For Random_share, Share_2 and Share_1 are employed,

i.e. if bit of Share_1 is 1, compliment of bit of Share_1is written into Random_share else bit of Share_2 is copied into Random_share.

For Key share, combination of Share_3 and Data_share are used, i.e. if bit of Data_share is 1, compliment of bit of Share_3 is written into Data_share2 else bit of Share_3 is copied as it is into Data_share2.

Step 5: XOR three shares to generate the key share which increases the complexity of retrieving the data at receiver side as:

```
            for i=1 to length(message)
             keyshare(i)=xor(Random_share(i),xor(Share_3
             (i),Data_share2(i)));
            end
```

Step 6: Transmit the key share and rest share.

On receiver side, while performing the decryption process, there is only need to XOR [17, 28] the key share and rest share to retrieve the original information (see Fig. 5 and 6).

Algorithm for Hierarchical Visual Cryptography (Receiver Side)

```
Step 1: Obtain the key share and rest share.
Step 2: XOR the bits of key share and rest share to
obtain the actual message.
        for i=1 to length(message)
            message(i)=xor(keyshare(i), restshare(i));
        end
Step 3: Actual decrypted data is available which is
finally decompressed to get secret data.
```

4.1 Steganography technique used

In the proposal, third layer of security is introduced by the steganography technique [25]. Here Improved BPCS (Bit Plane Complexity Segmentation) which is a spatial domain steganography technique is used to hide/embed the data into the cover image. The basis for improved BPCS is *Bit Plane Complexity Segmentation technique* (*BPCS*) which was first put forward by Kawaguchi and Eason [10]. This technique is chosen for its very special feature of high embedding capacity. In this scheme, data is embedded into noise like regions which cannot be differentiated by human eye [3]. Actually these noise like regions are those which has maximum possible changes in adjacent pixel values i.e. black and white pixels thus termed as complex regions. Under this algorithm firstly complexity of region is identified and then same amount of data is embedded into selected regions having high complexity, since complexity of data replacing image data is also high, therefore the picture quality is not degraded.

4.1.1 Proposed steganographic mechanism

The difference between BPCS and improved BPCS lies in the complexity module. This modification over the basic BPCS is in terms of complexity calculation. Complexity is defined as ratio of total number of pixel border where pixel transition is happening to total no of border in that block, it is denoted by C_b. For the block under consideration, C_b can be calculated using the mathematical formula:

$$C_b = \frac{k}{M}$$

where k is the no. of pixel border where transition is happening, and M is the total no of borders.

A parameter α is also used in finding threshold complexity as follows. It is calculated by using the chaotic map i.e. Logistic map equation, then the product of these two parameters

$$C_t = \alpha^* C_b$$

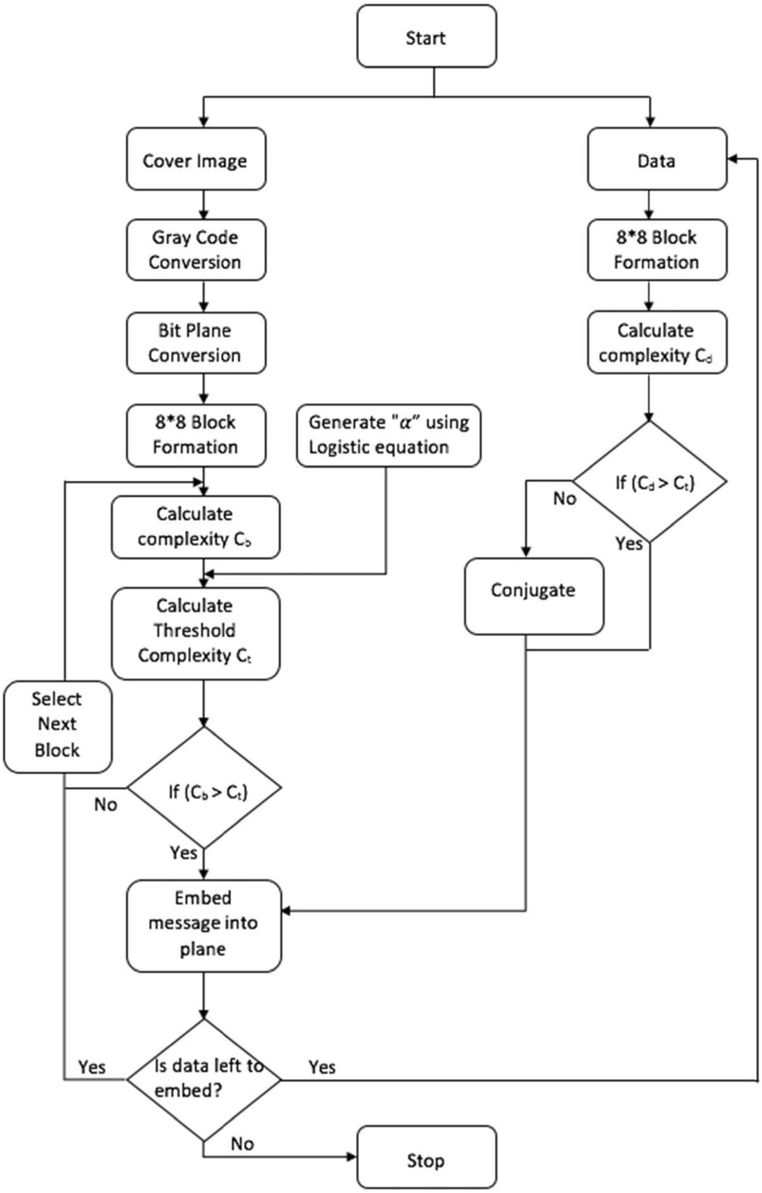


Fig. 6 Flow chart of Improved BPCS

C_t is called threshold complexity of the block. C_t is used to decide whether the block will be used for embedding or not. Complexity, C_b,of each 8x8 block is calculated, and if the complexity of a block is greater than threshold complexity C_t, then this block is used for embedding the data bits else it is skipped.

Here α is generated dynamically for every 8x8 block of cover image using chaotic map, so threshold complexity of every block is varying nature which induces high randomness and therefore ensures high security, hence the name Improved BPCS. Complexity of message blocks is also calculated, if it is less than threshold value, then conjugate of that block is taken to make it more complex and this information is stored in another 8x8 block named conjugate map.

The detailed flow chart for the improved BPCS steganography technique is given below

Before Explaining the flow chart in detail, few prerequisite processes are explained in detail to make explanation easier.

5 Canonical gray coding (CGC)

The idea behind this conversion is that in binary coded planes, there is a dilemma of 'Humming Cliff' in which two consecutive pixel values appear identical to human eye but differ greatly in bit representation. For example, pixel values 127 and 128 cannot be differentiated easily but their binary representation i.e. 01111111 and 10000000 differ completely. If secret data is embedded, there is a possibility that 01111111 can become 11111111 and 10000000 can become 00000000 which can highly change the picture resolution. But by using Canonical Gray Coding (CGC), this problem is resolved.

6 Conversion

Suppose $B_3B_2B_1B_0$ is binary data, then its gray code data $G_3G_2G_1G_0$ can be obtained as:

$G_{0=}B_0$
$G_1{=}G_0$ xor B_1
$G_2{=}G_1$ xor B_2
$G_3{=}G_2$ xor B_3

7 Bit plane slicing

Here the gray converted cover image is divided into bit planes by using bit plane slicing technique i.e. each pixel value is divided into 8 bits. Then an LSB plane i.e. bit 0 plane of same size is formed by combing all the LSBs of every pixel in that plane. Similarly,

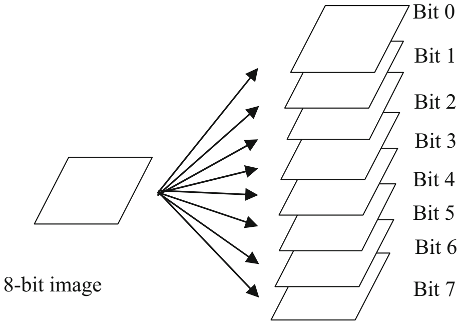


Fig. 7 Dividing of bits to form bit planes

for all other bits, planes are formed. Hence for an 8 bit pixel value, 8 planes are generated i.e. from bit 0 to bit 7 (see Fig. 7).

If the image is RGB i.e. colored image then eight bit planes are formed for each R, G and B planes. Therefore, total 24 bit planes are generated for a colored image. Now, each of the bit planes is divided into 8x8 size blocks. Then complexity of each block is calculated.

8 Conjugation

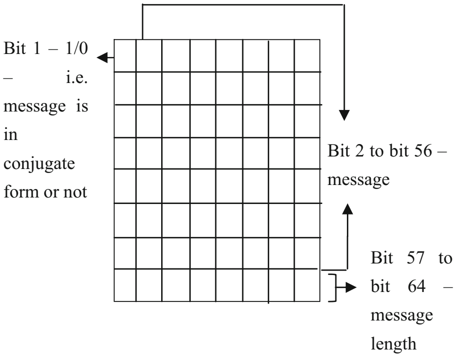
This process is used if text data which is getting embedded is not complex enough to embed. First text data is converted into form which is shown in Fig 8 and the it is xored with a chessboard pattern block of same size. Process is similar to the procedure used by Kawaguchi and Eason [10] in their Conjugation of binary image step.

From Fig. 8, first bit i.e. bit 1 of conjugate map contains the information about whether this block is also conjugated or not. And from second bit position i.e. from bit 2 to bit 56, conjugate status of message block is stored i.e. if the block is conjugated, 1 is stored at that place else 0 is stored. This process continues for each message bit block. The last row of this map block i.e. bit 57 to bit 64 contains information regarding the length or size of data. First conjugate map is embedded into the cover image and then message blocks so that while extracting the message blocks at the receiver side, their conjugate status and size is known in advance.

9 Explanation of flow chart

As shown in Fig. 6, first the pixel values are converted from binary to gray code. Thus this gray converted cover image is divided into bit planes by using bit plane slicing technique. Therefore, total 24 bit planes are generated for a colored image. Now, each of the bit planes is divided into 8x8 size blocks. Then complexity of each block is calculated denoted by C_b. A parameter α is also calculated by using the chaotic map, then the product of these two parameters, C_t is called threshold complexity. If the complexity of a block is greater than threshold complexity, then this block is used for embedding the data bits else it is skipped. Complexity of message blocks is also calculated, if it is less than threshold value, then conjugate of that block is taken to

Fig. 8 Structure of conjugate map


Bit 1 – 1/0 – i.e. message is in conjugate form or not
Bit 2 to bit 56 – message
Bit 57 to bit 64 – message length

make it more complex and this information is stored in another 8x8 block named conjugate map.

In this technique, first a basic threshold for each bit plane is set, then some incremental value is added in it to make it dynamic threshold. This threshold is not constant even in the same bit plane.

Let x be the data generated using chaotic map, where $0 \le x \le 1$, then the corresponding increments are as follows:

-2 for $0 \le x < 0.2$
-1 for $0.2 \le x < 0.4$
0 for $0.4 \le x < 0.6$
1 for $0.6 \le x < 0.8$
2 for $0.8 \le x \le 1$

Algorithm for Improved BPCS Steganography (Sender Side)
Step 1: Convert the message into binary form and then in 8x8 size blocks.
Step 2: Obtain the cover image and then apply Canonical Gary Coding (CGC) on it after converting the pixel values into binary form.
Step 3: Apply bit plane slicing technique to convert the gray coded planes into bit planes i.e. to generate 8 planes for an 8-bit pixel.
Step 4: These bit planes are further converted into 8x8 block planes, and then maximum complexity C_b is calculated.
Step 5: Another parameter α is generated dynamically using chaotic map. Using these two parameters complexity threshold, C_t is calculated.
Step 6: This complexity threshold, C_t is compared with the complexity value of each bit plane.
Step 7: If the calculated complexity value of given bit plane is greater than the required one i.e. C_t, the message is embed into it else complexity of next plane will get compared.
Step 8: Before embedding the message, its complexity is also checked using the same procedure. If the message is complex, then it is embedded else its conjugate is used.
Step 9: Repeat steps 6 to 8 for each bit plane and message block, till message limit is reached.
Step 10: Convert the 8x8 bit blocks into bit planes and then reform the image to get the stego image.
Step 11: Transmit the stego image.

At the receiver side too, the same process is applied on the stego image to retrieve the message i.e. stego image is again deformed into bit planes and then into 8x8 bit blocks. Then calculation of complexity on the basis of which first conjugation map is extracted which provides the information about amount and conjugate status of message blocks.

Algorithm for Improved BPCS Steganography (Receiver Side)
Step 1: Obtain the stego image.
Step 2: Apply Canonical Gary Coding (CGC) on it after converting the pixel values into binary form.
Step 3: Apply bit plane slicing technique to convert the gray coded planes into bit planes i.e. to generate 8 planes for an 8-bit pixel.
Step 4: These bit planes are further converted into 8x8 block planes, and then maximum complexity C_b is calculated.
Step 5: Another parameter α is generated dynamically using chaotic map. Using these two parameters complexity threshold, C_t is calculated.

Step 6: This complexity threshold, C_t is compared with the complexity value of each bit plane.
Step 7: If the calculated complexity value of given bit plane block is greater than the required one i.e. C_t, the conjugate map is extracted first from this block from which status of conjugate message blocks will be checked else skip that block.
Step 8: Extract the length of message from last row, and finally retrieve the message from bit locations 2–56.
Step 9: Repeat steps 7 and 8 for all the bit planes as per the length of the message.
Step 10: Now, recover the secret message by converting the message obtained in step 8 into original form.

10 Simulation setup parameters and performance metrics

10.1 Simulation Set up parameters

The experiments are carried out on a personal computer. Table 2 provides the specifications and set up parameters.

For the purpose of comparison, four papers are implemented in MATLAB and their results are compared with the proposed scheme.

References	Scheme Name
Ref- [1]	Divya Chaudhary et al.
Ref- [7]	Md. Rashedul et al.
Ref- [13]	PiyushMarwaha et al.
Ref- [23]	Peipei Shi et al.

10.2 Performance metrics

For complete analysis of proposed technique, various parameters have been observed, which are summarized into following categories:

10.2.1 Encrypted code analysis

This is the very first test for any cryptography scheme. It is performed to measure avalanche effect on the proposed scheme. Avalanche effect means a small change in plaintext should create a significant change in cipher text.

10.2.2 Key space analysis

Encryption should be very sensitive for a slight change in the value of key, as this will create a huge change in the encrypted output. Using a large key space ensures resistance of technique towards brute force attacks i.e. breaking of algorithm by trial and error method to get the key by using automated software to generate a large number of consecutive guesses. Larger the key space, lower the possibility of this attack. So, for a scheme to be successful, key size should be large to get a large key space.

Table 2 Set up Parameters

Specifications	
Processor	Core-i3 1.7GHz RAM 4GB
Image size	64*64 128*128 256*256 512*512 1024*1024
Image type	.jpg
Simulation tool	MATLAB 7.14.0.739 64 bit (win 64)
Software version	2010
Text used for embedding	“ABCDEFABCDEFABCD”
Huffman encoding	Number of symbols n=26 Alphabets used are ‘A to Z’
Hierarchical visual cryptography	Number of share in which data is divided=4
Threshold parameter α used in BPCS technique	
k1	0.5
k2	3.6
k3	0.51
k4	3.7
k5	0.52
k6	3.8

10.2.3 Similarity analysis

Similarity analysis is used to measure the closeness between input i.e. cover image and stego image.

10.2.4 Correlation coefficient

This parameter is a measure of the linear correlation i.e. dependence between two images. Its range is between −1 to +1 both inclusive, where 1 signifies perfect match and −1 signifies total mismatch. The correlation coefficient can be calculated as:

$$r\,A\,B = \frac{cov(A,B)}{\sqrt{D(A)}\sqrt{D(B)}}$$

$$cov(A,B) = \frac{1}{N}\sum_{i=1}^{N}(Ai\text{–}E(A))(Bi\text{–}E(B))\Big)$$

$$D\quad(A)\quad = \quad \frac{1}{N}\sum_{i=1}^{N}(Ai\text{–}E(A))^2$$

$$D\quad(B)\quad = \quad \frac{1}{N}\sum_{i=1}^{N}(Bi\text{–}E(B))^2$$

Where E (*A*)= mean of *A* and E (*B*)= mean of B, *A* and B denote two images for which correlation needs to be calculated, and N is the total number of elements obtained from the data.

10.2.5 Universal image quality index (UIQI)

In an image, pixels values available at different positions shows different effect on Human Visual System (HVS). If some distortion or changes is introduced in the image, such distortion in image is calculated as a combination of three factors loss of correlation, contrast distortion and luminance distortion.

$$Luminance\ distortion,\ L(A,B)=\frac{2\ \mu_A\mu_B}{{\mu_A}^2+{\mu_B}^2}$$

$$Contrast\ distortion,\ C(A,B)=\frac{2\sigma_A\sigma_B}{{\sigma_A}^2+{\sigma_B}^2}$$

$$Loss\ of\ correlation,\ S(A,B)=\frac{2\sigma_{AB}}{\sigma_A+\sigma_B}$$

$$UIQI(A,B)=L(A,B)*C(A,B)*S(A,B)$$

Where A is cover image, μ_A and σ_A are mean and standard deviation, respectively of A. B is stego image, μ_B and σ_B is mean and standard deviation, respectively of B. σ_{AB} is covariance between A and B.

10.2.6 Robustness analysis

Robustness analysis is a measure of preservence of picture quality of the stego image in the presence of noise.

10.2.7 Peak signal noise ratio (PSNR)

It is defined as the ratio between the maximum possible power of a signal and the power of corrupting noise. It is the ratio of peak square value of pixels by mean square error (MSE). It is expressed in decibel (db). The PSNR is defined as:

$$\mathrm{PSNR}=10.\log_{10}\left(\frac{{\mathrm{MAX_I}}^2}{\mathrm{MSE}}\right)$$

where,

$\mathrm{MAX_I}$ represents maximum value of pixel of the image
MSE is the mean square error.

10.2.8 Embedding capacity analysis

Embedding capacity can be defined as the ratio of number of bits that can be embedded to the total number of bits. This analysis is basically used to test whether a technique is capable to embed a large amount of data or not.

$$\text{Embedding Capacity}=\frac{\text{Number of bits that can be embedded}}{\text{Total number of bits}}$$

10.2.9 Information Entropy Analysis

Information entropy is the measure of amount of randomness. The entropy H(S) of a message m can be calculated as

$$H(S) = \sum_{i=0}^{n-1} P(S_i) log_2 \frac{1}{P(S_i)}$$

Where P (Si) signifies the probability symbol Si, log is of base 2. If there are 256 possible outcomes of the message S with equal probability, then H(S) = 8, which is an ideal value for this case. The value of entropy close to value 8 signifies that encrypted output is highly random in nature.

11 Results

11.1 Snapshots

Comparison of various images after applying the different approaches is given in Fig. 9.

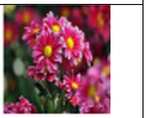
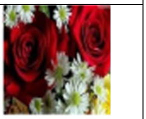
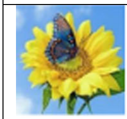
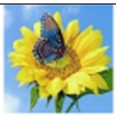
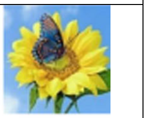
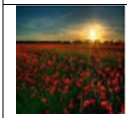
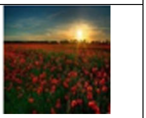
Original image	Image after applying Peipei Shi et. al. method	Image after applying PiyushMarwaha et.al. method	Image after applying Md. Rashedul et.al. method	Image after applying Divya Chaudhary et. al. Method	Image after applying Proposed technique
					
					
					
					

Fig. 9 Snapshots

Table 3 Code Assessment

Cryptography Technique	Original Secret 1	Original Secret 2	Encrypted Original Secret 1 length	Encrypted Original Secret 2 length	Avalanche effect(Bits Change)	Avalanche effect(% Bits Change)	Length Varying	Complexity
Proposed technique	ABCDEFABCDEFABCD	BBCDEFABCDEFABCD	44 bits	43 bits	27 bits	62.7 %	Yes (Data length changed from 44 bit to 43 bits)	(M)
Divya Chaudhary et al.	ABCDEFABCDEFABCD	BBCDEFABCDEFABCD	88 bits	86 bits	36 bits	40.9 %	Yes (Data length changed from 88 bit to 86 bits)	(M)
Md. Rashedul et al.	ABCDEFABCDEFABCD	BBCDEFABCDEFABCD	128 bits	128 bits	60 bits change	46 %	No	(M)
PiyushMar-waha et al.	ABCDEFABCDEFABCD	BBCDEFABCDEFABCD	128 bits	128 bits	32 bits change	25 %	No	(M)

Table 4 Key Space Analysis

Techniques	Proposed technique	Divya Chaudhary et al.	Md. Rashedul et al.	PiyushMarwaha et al.	Peipei Shi et al.
Key size	Same as data length(say l)	Same as data length(say l)	128 bits	64 bits	No key
Key space	2^l	2^l	2^{128}	2^{64}	–

As per results of visual analysis, it is not possible to identify the presence of any type of information in the image. Both images are seems to be alike. Results for this parameter are comparable for all other techniques.

11.2 Encrypted code analysis

After analyzing Table 3, it can be observed that for proposed scheme and Divya Chaudhary et al., length of the cipher is varying, because of Huffman compression, which is an additional advantage as for any change along with data value, data length is also changing. From table it can be seen that only for proposed scheme, with the small change in input, maximum percentage change in output can be seen (27 bits out of 44 bits). It can also be seen that complexity of all the schemes depend upon the length of data to be embedded (M). For the proposed scheme, due to compression, amount of data to embed is very less, as compared to other schemes, therefore have less complexity.

11.3 Key space analysis

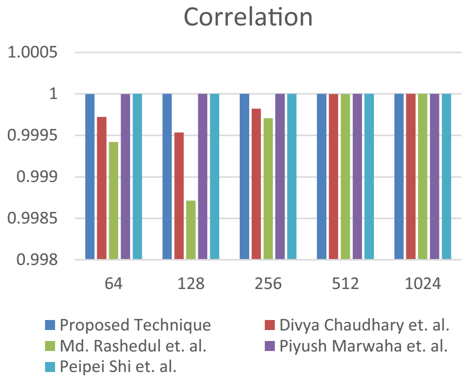
A good encryption scheme should have a large key space as it is directly related with brute force attack. As the key size increases, possibility of this attack decreases. Table 4 depicts that key size for the proposed technique and Divya Chaudhary et al.[1] is not fixed, it is varying with data length. For a large amount of data, the key size will be very large and hence the key space which ultimately increases the resistance to brute force attacks. This result makes this technique more secure for bigger size of secret data.

11.4 Similarity analysis

11.4.1 Correlation coefficient

Figure 10 shows that correlation coefficient, which is a measure of similarity between input image and the stego image, provides very good results for proposed technique, PiyushMarwaha et al.[13] and Peipei Shi et al.[23] as the value of correlation coefficient for these schemes remains 1, which is an ideal value, for all pixel values. While for other schemes, this value is less than 1 and rises with the pixel size.

Fig. 10 Effect of pixel size on correlation coefficient

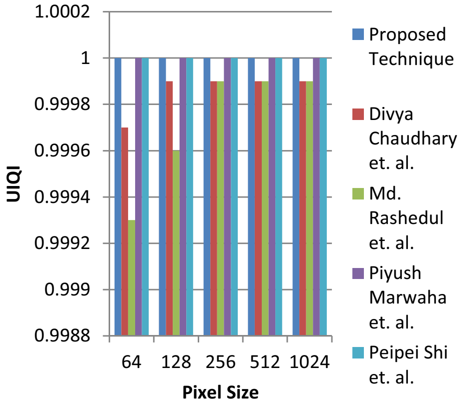


11.4.2 Universal image quality index (UIQI)

Figure 11 depicts that Universal Image Quality Index (UIQI), which is a measure of image quality in terms of contrast, luminance and loss factor, provides very good results for proposed technique, PiyushMarwaha et al.[13] and Peipei Shi et al.[23] as the value of quality index for these schemes remains 1, which is an ideal value, for all pixel values. While for other schemes, this value comes very near to 1 with an increase in the pixel size.

With the analysis of results of these two parameters and snapshots we can conclude that proposed technique don't provide any indication regarding presence of information into cover image. Although these results are very similar with other hybrid mechanisms, but main strength of proposed technique lies in quantity of data to be transmitted.

Fig. 11 Effect of pixel size on Universal Image Quality Index (UIQI)



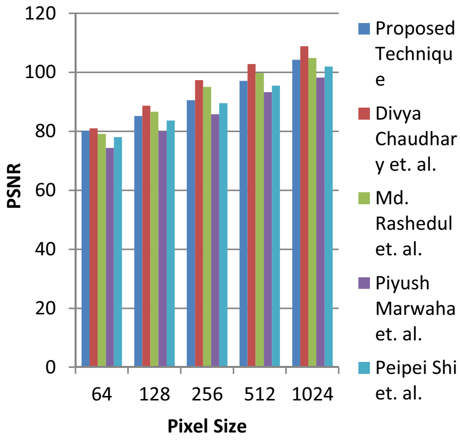


Fig. 12 Effect of pixel size on Peak Signal to Noise Ratio (PSNR)

11.5 Robustness analysis

11.5.1 Peak signal noise ratio (PSNR)

Figure 12 shows that PSNR, which is a measure of picture quality of stego image, is highest for Divya Chaudhary et al.[1] and is least for PiyushMarwaha et al.[13]. Higher the PSNR value, higher the picture quality. The proposed technique has low PSNR value. The same can be confirmed from Table 5 and Fig. 12, but it is higher than Peipei Shi et al.[23], which is the basis for proposed technique. Hence, the proposed technique proves to be better than standalone Improved BPCS in terms of PSNR.

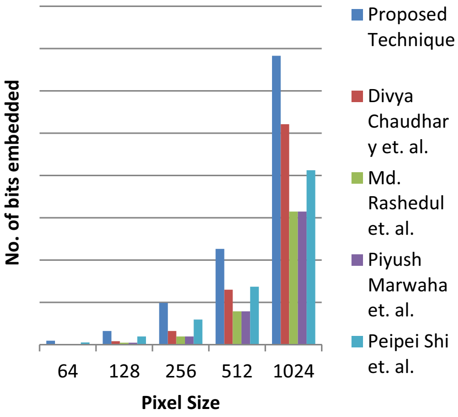
11.6 Embedding capacity analysis

Figure 13 depicts that embedding capacity, which is a measure of number of data bits that can be embedded into cover image, provides best result for the proposed scheme

Table 5 PSNR Analysis

	Proposed Technique	Divya Chaudhary et al.	Md. Rashedul et al.	PiyushMarwaha et al.	Peipei Shi et al.
64	80.17954982	81.00929173	79.1728482	74.40163566	77.999
128	85.19344812	88.67799457	86.59235149	79.99471577	83.6896
256	90.52323884	97.38704761	95.04621555	85.75202629	89.5033
512	97.08741538	102.7876658	99.76345467	93.28530296	95.5239
1024	104.2234568	108.848328	104.8689279	98.19439835	101.987654

Fig. 13 Effect of pixel size on embedding capacity



as this value is highly growing with the pixel size. There is no other scheme (see Table 6) which can provide this much enhancement in this property as all other techniques have very less growth in embedding capacity as compared to the proposed one.

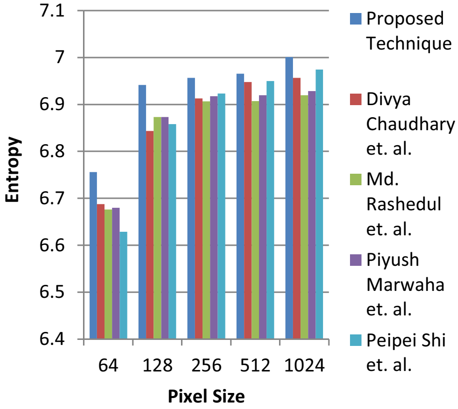
11.7 Information entropy analysis

From Fig. 14, it is observed that entropy, which is a measure of randomness of data, provides best result for the proposed technique as it is close to ideal value 8. It is due to the usage of chaotic map which has been used to generate the random values. Usage of this map raises the randomness which in turn increases the entropy. Higher the entropy, higher the unpredictability of data and hence, ensures security. All other schemes have lower entropy as compared to proposed technique which results in lack in complexity and hence, in security.

Table 6 Embedding Capacity Analysis

Image	Proposed Technique	Divya Chaudhary et al.	Md. Rashedul et al.	PiyushMarwaha et al.	Peipei Shi et al.
64	93810	20352	12288	12288	56640
128	322240	81408	49152	49152	194560
256	989404	325632	196608	196608	597376
512	2267552	1302528	786432	786432	1369088
1024	6827036	5210112	3145728	3145728	4121984

Fig. 14 Effect of entropy on pixel size



12 Conclusion

This paper is an effort to increase the embedding capacity as well as entropy of a hybrid security system while preserving picture quality. Table 7 depicts the overall comparison of various security mechanisms present in the literature with the proposed one. It is clear from the key space analysis that proposed scheme has large key space and it is of varying nature depending upon length of data thereby improving the security. From correlation analysis and UIQI analysis, we can conclude that proposed scheme has one of the highest correlation and UIQI coefficients thereby depicting there is minimum change in image quality. In terms of embedding capacity, proposed scheme has the highest embedding capacity among all compared and has shown highest entropy results.

Table 7 Overall Comparison

Parameters	Divya Chaudhary et al.	Md. Rashedul et al.	Piyush Marwaha et al.	Peipei Shi et al.	Proposed technique
Encrypted Code	Unreadable and compressed	Unreadable	Unreadable	Unreadable	Unreadable and compressed
Key Space	Large and varying	Small and fixed	Small and fixed	No key space as there is no key	Large and varying
Correlation Coefficient	Higher	High	Highest	Highest	Highest
UIQI	Higher	High	Highest	Highest	Highest
PSNR	Highest	Higher	Lowest	Low	High
Embedding Capacity	Higher	Low	Low	High	Highest
Entropy	Higher	Low	Low	High	Highest

References

1. Chaudhary D, Gupta S, Kumari M (2015) “A Novel Hybid Security mechanism for Data Communication Networks” accepted for publication in Inderscience Journal
2. Chavan PV, Atique M (2012) “Design of Hierarchical Vuisual Cryptography”, in proceedings of International Conference on Engineering, pp 1–3
3. Goel S, Rana A, Kaur M (2013) “A Review of Comparison Techniques of Image Steganography” in proceedings of Global Journal of Computer Science and Technology Graphics & Vision, Vol. 13, Issue 4
4. Gokul M, Umeshbabu R, Shriram K (2012) Vasudevan, Deepak karthik, “hybrid steganography using visual cryptography and LSB encryption method. Int J Comput Appl 59:5–8
5. Hodeish ME, Dr Humbe VT (2014) State-of-the-Art visual cryptography schemes. Int J Elect Commun Comput Eng 5(2):412–420
6. Huffman DA (1952) A method for the construction of minimum redundancy codes. IRE 40(9):1098–1101
7. Islam RMd, Siddiqa A, Uddin MdP, Mandal AK and Hossain MdD (2014) “An Efficient Filtering Based Approach Improving LSB Image Steganography using Status Bit along with AES Cryptography” in proceedings of 3rd International Conference On Informatics, Electronics & Vision, pp 1–6
8. Jain Y, Sharma G, Anand G, Dhall S, “A Hybrid Security Mechanism based on DCT and Visual Cryptography for Data Communication Networks” accepted for publication in CSI-2015 Journal
9. Kaspersky Lab and INTERPOL (2014) “Mobile cyber-threats”, pp 11–12
10. Kawaguchi E, Eason RO (1998) Principle and applications of BPCS- steganography. Soc Photog Instrumentation Eng (SPIE) 3528:464–473
11. Lokesh K (2012) Novel security scheme for image steganography using cryptography technique. Int J Ad Res Comput Scie Soft Eng (IJARCSSE) 2:143–146
12. Mandal S, Das S, Nath A (2014) Data hiding and retrival using visual cryptography. Int J Innov Res Ad Eng 1(1):102–110
13. Marwaha P, Marwaha P (2010) Visual Cryptographic Steganography in Images” in proceedings of Second International conference on Computing, Communication and Networking Technologies, pp 1–6
14. Masud Karim SM, Rahman Md S, Hossain Md. I (2011) “A New Approach for LSB Based Image Steganography using Secret Key”, in proceedings of 14th International Conference on Computer and Information Technology (ICCIT 2011), pp 22–24
15. Mohammad AA, and Abdel F (2012) “Public-Key Steganography Based on Matching Method” in European Journal of Scientific Research, pp 223–231
16. Nivedhita R, Dr Meyyappan T (2012) Image security using steganography and cryptographic techniques. Int J Eng Trends and Technol 3:366–371
17. Pallavi Vijay C, Dr Mohammad A, Dr Latesh M (2014) Design and implementation of hierarchical visual cryptography with expansionless shares. Int J Net Secur Appl 6(1):91–102
18. Pye Pye A, Tun Min N (2014) A novel secure combination technique of steganography and cryptography. Int J Inf Technol /Mod Comput (IJITMC) 2:55–62
19. Ramakrishna M, Veera Raghava Rao A, Dr Srinivasan Kumar D (2012) Securing information: cryptography and steganography. Int J Comput Scie Inf Technol 3:4251–4255
20. Schwartz ES (1964) An optimum encoding with minimum longest code and total number of digits. Inf Control 7:37–44
21. Shailender G, Ankur G, Bharat B (2012) Information hiding using least significant Bit steganography and cryptography. J Mode Educ Comput Scie 6:27–34
22. Shelke FM, Dongre AA, Soni PD (2014) Comparison of different techniques for Steganography in images. Int J App Innov Eng Manag 3(2):171–176
23. Shi P, Li Z (2010) “An improved BPCS Steganography based on Dynamic Threshold”, in proceedings of International Conference on Multimedia Information Networking and Security, pp 388–391
24. Shingote Parshuram N, Syed Akhter H, BhujpalPallavi M (2014) Advanced security using cryptography and LSB matching steganography. Int J Comput Elec Res 3:52–55
25. Singh S, Siddiqui TJ (2012)“A Security enhanced Robust Steganography Algorithm for Data Hiding” in proceedings of International Journal of Computer Science Issues, Vol. 9, Issue 3, No. 1, pp 131–139
26. Stallings W, “Cryptography and Network Security: Principles and Practice” 4th edition.

27. Usha S, SathishKumal GA, Boopathybagan K (2011) A secure triple level encryption method using cryptography and steganography. Int Conf Comput Scie Net Technol 2:1017–1020
28. Vinish A, Dr Senthil Prakash T, Ajmal H (2015) Enhanced hierarchical design for visual cryptography-overview. Int J Eng Technol Scie 2(4):46–50

Miss Neha Tayal is a B.tech(Electronics and communication), pursuing M.Tech in the field of Electronics and communication at YMCA University of Science and Technology, Faridabad. Her research interest includes Encryption, Steganography, Image analysis.

Mr. Ritesh Bansal is a B. tech in Electronics Instrumentation & Control Engineering from YMCA University of Science & Technology, Faridabad, India. His research interest includes work Encryption, Digital Image processing, Steganography, Network Security.

Mr. Sangeeta Dhall is B.tech (I&C), M.Tech (EI). Her academic interests include Embedded System, Image Processing. Currently working as Assistant Professor in Electronics Engineering department at YMCA University of Science and Technology, Faridabad, India.

Mr. Shailender Gupta is B.Tech (Electronics Engineering), M.Tech (Computer Engineering) and recieved his Ph. D in the area of ad-hoc mobile network security. His academic interests include network security, Signal Processing, automata theory and fuzzy logic. Currently working as Assistant Professor in Electronics Engineering department at YMCA University of Science and Technology, Faridabad, India.