A NOVEL IMAGE ENCRYPTION SCHEME WITH HUFFMAN ENCODING AND STEGANOGRAPHY TECHNIQUE

Manju Kumari[1], Vipin Pawar[2] and Pawan Kumar[3]

[1]YMCA University of Science and Technology, Faridabad, India
[2]Deenbandhu Chotu Ram University of Science and Technology, Murthal, India
[3]Department of Mathematics, Maitreyi College, University of Delhi, Chanakya puri, New Delhi-110021, India

ABSTRACT

In today's day and age when everything is done with the aid of computing technology, the need for confidential communication has drastically increased. Not only the sensitive data such as top intelligent secrets of our nation but personal information of common people needs to be secure. Several combinations of cryptography and steganography techniques in different ways are used by researchers over the past to protect the data being transmitted. Cryptography uses mathematical algorithms to convert the data into an incomprehensible form and Steganography, on the other hand hides the data in a carrier such as image, data, audio or video. Cryptography provides necessary mechanisms for providing accountability, accuracy and confidentiality in public communication mediums such as the Internet and steganography is used in other fields such as copyright, preventing e-document forging etc. We are of the opinion that this security mechanism can further be increased by incorporating the use of Huffman coding in order to reduce the data length. This paper is an effort in the direction to hide, secure and compress the data. It explains the executed procedure by applying various encryption techniques one by one and our aim is to get the best security out of the existing ones. The proposed technique is implemented in MATLAB2016a and the results shown in this paper that our technique is better approach then the conventional techniques.

KEYWORDS

Cryptography; Steganography; Huffman Coding; Data Compression.

1. INTRODUCTION

There are many techniques available to provide security in the communication field over the network such as cryptography and steganography. Cryptography[3,4] is a Greek word that means "the art of writing secrets". Basically, cryptography transforms the information into unreadable form that is not understood by anyone except the intended recipients, who is only able to retrieve the original data using a secret key. Cryptographic algorithms results into transformed data from original data by using a secret key is also called Ciphers and the technique is called encryption. Special programs are designed to protect sensitive information on public communication networks. During encryption, ciphers transform the original plaintext message into unpredictable ciphertext. Decryption is the reverse process of encryption in which plaintext is retrieved back from the ciphertext. In this paper some of the best encryption schemes are discussed with the help of block diagrams like DES, Triple-DES, BLOWFISH, AES, RC4[20].

The main goal of Steganography[1,9,11] is to hide information under the other media called a cover so that the person other than the receiver will not notice the presence of the information. Steganography is also a Greek word which means covered writing and essentially it means "to

hide in plain sight". This is a major difference in this method and the other methods of exchange of information like cryptography because the individuals notice the presence of secret information by seeing the coded message but they will not be able to comprehend the data. However, in steganography, the existence of the information will not be noticed at all due to the cover media. Steganography[12,14] should hide information well enough such that the unintended recipients do not suspect the medium contains any hidden data. Most of the steganography technique works generally have been carried out on images, video clips, texts, music and sounds. Hence using a combination of steganography and cryptography technique offers the improved information security mechanism.

To provide more efficiency to the proposed hybrid technique, further data compression technique is used to enhance the speed of data transmission over the network[18]. For this purpose Huffman encoding technique is employed with the combination of cryptography and steganography techniques. Huffman encoding[2] is a method to code a sequence of data items with minimum number of bits necessary and provides lossless data compression. It is a way to assign varying number of binary codes to data symbols such that the overall number of bits used to encode a normal string of those symbols get reduced. Firstly we encrypt the plaintext using one of the studied cryptography techniques; the ciphertext produced is then compressed using Huffman encoding procedure. The ciphertext completely changes after applying Huffman encoding due to data compression and this data is then embedded in the image using steganography techniques thus providing better security.

The rest of the paper will help in the better understanding of the procedure which is organising as follows: Second section consists specific features of the studied techniques thorough literature survey. Third section elaborates the proposed scheme and fourth section provides all details of simulation setup parameters. Fifth section is enclosed with simulation results and finally, the research is concluded with future scope in sixth section.

2. LITERATURE SURVEY

In this section firstly various cryptography schemes[15] has been discussed with best of their features in a tabular format as follows:

Table 1: Description of Cryptography Techniques

Proposed by	**Year of Publication**	**Applied Encryption Technique**	**Advantages**	**Disadvantages**
S.Fluthrer, et. al. [5]	2001	RC4	Low execution time High entropy Low correlation High pixel and key – sensitivity	Moderate PSNR
M. Zeghid, et. al. [6]	2007	AES	High entropy High security	Very large execution time Very low correlation value Low key space
Said.El Zoghdy, et. al. [7]	2011	DES	High entropy High PSNR	Large execution time High correlation value Low key space
Quist-Aphetsi Kester, et. al. [10]	2012	Vigenère Ciphere	Less execution time Moderate entropy Moderate PSNR	High correlation value Low pixel sensitivity Low key sensitivity Low key space Easily predictable key

Some other techniques based on steganography and cryptography have also studied to enhance this research. One of the papers[17] is written by Shailender Gupta, proposed a hybrid model that combines steganography and cryptography. The Cryptographic technique used was RSA and Diffie Hellman and the steganographic technique used was LSB substitution. Since the public key mechanisms have higher time complexity than other mechanisms, Thus it is not practical to use them for sharing large data.

Another paper written by Surbhi Singhania[22] explains the combination of cryptography and steganography. The techniques used for cryptography are DES and RC4 and steganography is done by using Substitution, Distortion and Direct Cosine Transform. The benefit of the combination is that even if the antagonist is able to detect the confidential information, he still needs the decryption key to decrypt the data.

Sujay Narayana[16] proposed a scheme using S-DES algorithm combined with LSB substitution technique. The image to be hidden was first encrypted using a key and the encrypted image was embedded in the cover image. Their work was primarily focused on the change in image quality. Further to reduce the data length Huffman encoding[2] has been used. In Huffman coding, shorter codes are assigned to the most frequently occurring symbols and longer codes are assigned to lesser probable symbol in a particular document. The procedures in detail are defined in the upcoming section.

3. THE PROPOSED SCHEME

3.1. Sender Side Procedure

The original data file containing plaintext is read in the form of ASCII characters. This plaintext is then converted into ciphertext by applying cryptographic algorithms such as DES, RC4 and Vigenere Square. The ciphertext formed is encoded by using Huffman coding technique of data compression. The compressed ciphertext is embedded in the image using steganographic techniques such as LSB substitution and Distortion substitution.

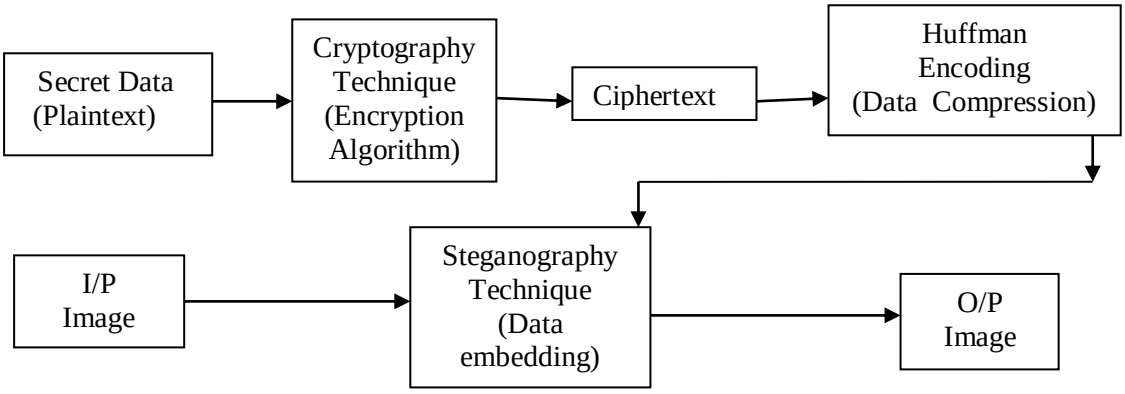

Figure 1. Mechanism at sender side

3.1.1. Cryptography Techniques Used

a) Data Encryption Standard (DES): DES[7,8] is a 64-bit block cipher which means that it encrypts data 64 bits at a time. The indifference to a stream cipher in which only one bit at a time is encrypted. DES is based on a cipher known as the Feistel block cipher. It consists of a number

of rounds where each round contains bit-shuffling, non-linear substitutions (S-boxes) and exclusive OR operations. DES takes two inputs - the plaintext and the secret key. DES is a symmetric, 64-bit block cipher as it uses the same key for both encryption and decryption and only functions on 64-bit blocks of data at a time. The actual key size used is 56 bits, though a 64-bit (or eight-byte) key is actually input. The least significant bit of each byte is either used for parity or set arbitrarily and does not raise the security in any way. All blocks are numbered from left to right which makes the 8-bit of each byte the parity bit. Once a plain-text message is received for encryption, it is arranged into 64-bit blocks required for input. If the number of bits in the message is not evenly divisible by 64, then the last block will be padded with extra bits. Multiple permutations and substitutions are included throughout in order to increase the difficulty in performing cryptanalysis on the cipher.

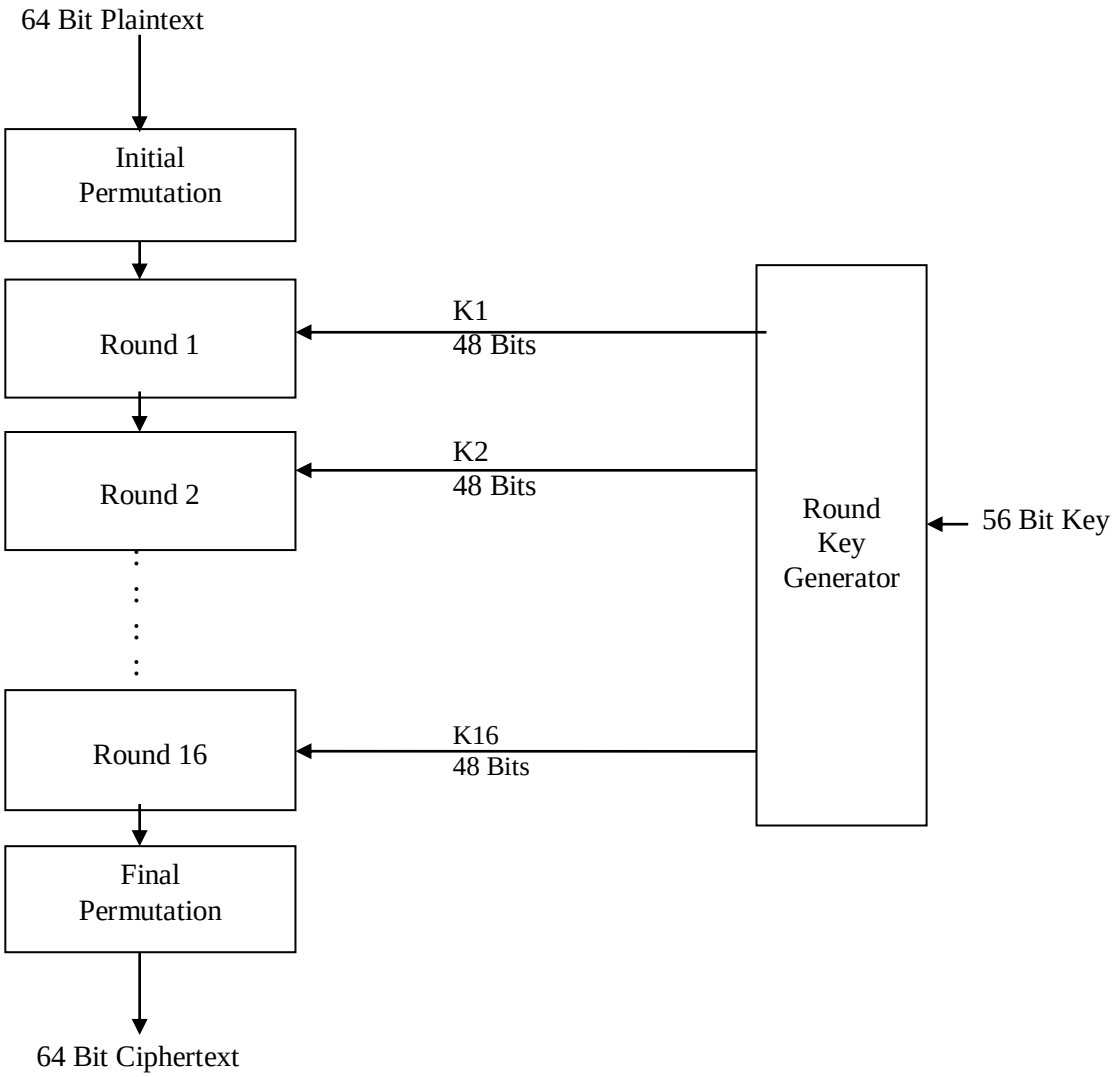

Figure 2. Flow Diagram of DES Algorithm

Figure 2. Shows the series of events that occur during an encryption operation. DES performs an initial permutation on the entire 64-bit block of data. It is then split into 2, 32-bit sub-blocks, Li and Ri which are then passed into what is called as round[21], which are 16 (the subscript is in Li and Ri specify the current round). Each of the rounds is identical and the effect of increasing their number is to double the algorithm's security. For DES the number chosen was 16, probably to

guarantee the elimination of any connection between the ciphertext and either the plaintext or key. At the end of the 16th round, the 32 bit Li and Ri output quantities are swapped to create what is known as the pre-output. This [R16, L16] concatenation is permuted using a function which is the exact opposite of the initial permutation. The output of this final permutation is the 64-bit ciphertext.

Details of individual rounds: Details of an individual round can be seen in figure 3. The main operations on the data are included into what is referred to the cipher function and labelled as **F**. This function accepts two different length inputs of 32-bits and 48-bits and output is a single 32-bit number. Both the data and key are operated on in parallel; however the operations are quite different. The 56-bit key is split into two 28 bit halves Ci and Di (C and D being chosen so as not to be puzzled with L and R). The value of the key used in any round is simply a left cyclic shift and a permuted contraction of that used in the previous round.

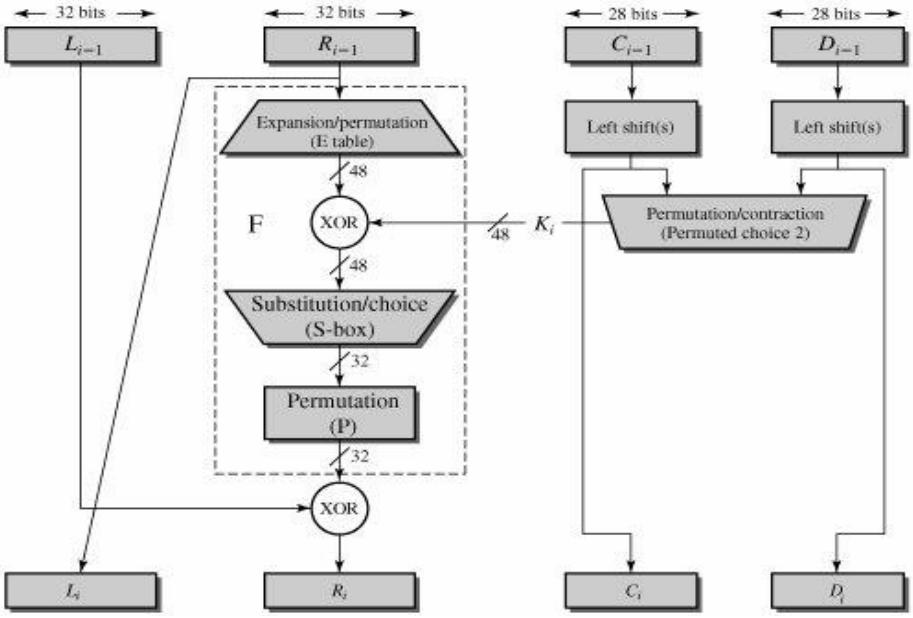

Figure 3. Single DES Round

The function **F** is the main part of every round and consists of four separate stages:

1) The E-box expansion permutation - The 32-bit input data from Ri−1 is expanded and permuted to give the 48-bit data necessary for combination with the 48-bit key. The E-box expansion permutation delivers a larger output by splitting its input into 8, 4-bit blocks and copying every first and fourth bit in each block into the output in a defined manner. The security offered by this operation comes from one bit affecting two substitutions in the S-boxes.

2) The bit by bit addition modulo 2 (or exclusive OR) of the E-box output and 48-bit subkey K_i

3) The S-box substitution - This is a highly significant substitution which accepts a 48-bit input and outputs a 32-bit number. The input to the S-boxes is 48 bits long arranged into 8, 6-bit blocks (b1, b2, . . . , b6). There are 8 S-boxes (S1, S2, . . . , S8) each of which accepts one of the 6-bit blocks. The output of each S-box is a 4-bit number. Each of the S-boxes can be thought of as a 4 × 16 matrix. Each cell of the matrix is identified by a coordinate pair (i, j), where i is 0 to 3 and j is 0 to 15. The value of i is taken as the decimal representation of the first and last bits of the input to each S-box, i.e. i = Dec (b1b6) and the value of j is take from the decimal representation

of the inner remaining four bits, i.e. j = Dec (b2b3b4b5). Each cell within the S-box matrices contains a 4-bit number which is output once that particular cell is selected by the input.

4) The P-box permutation - This simply permutes the output of the S-box without changing the size of the data. It is simply a permutation and nothing else. It has a one to one mapping of its input to its output giving a 32-bit output from a 32-bit input.

b) RC4 : RC4[3,5] is a stream cipher designed in 1987 by Ron Rivest for RSA Security. It is a variable key size stream cipher with byte-oriented operations. The algorithm is based on the use of a random permutation. A variable-length key from 1 to 256 bytes is used to initialize a vector S, with elements S[0], S[1], …, S[255]. S contains a permutation of all 8-bit numbers from 0 through 255 at all times. For encryption and decryption, a byte k is generated from S by selecting one of the 255 entries in a systematic fashion. As each value of k is generated, the entries in S are once again permuted. The procedure for RC4 encryption and decryption are comprehended by figure 4.

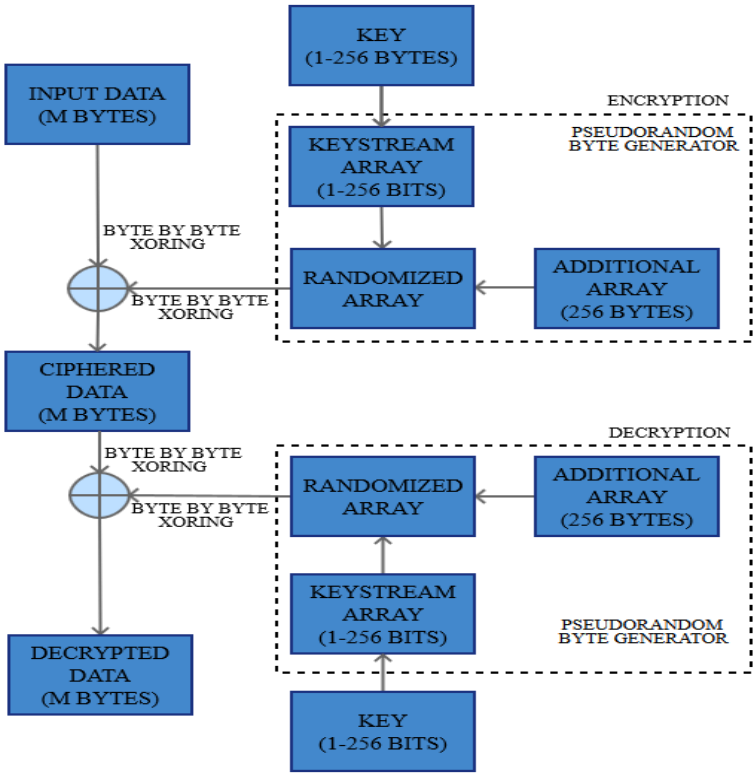

Figure 4. RC4 Encryption Scheme

Initialization of S: To begin, the entries of S are set equal to the values from 0 to 255 in ascending order; that is; S[0] = 0, S[1] = 1, …, S[255] = 255. A temporary vector, T, is also created. If the length of the key K is 256 bytes, then K is transferred to T. Otherwise, for a key of length keylen bytes, the first keylen elements of T are copied from K and then K is repeated as many times as necessary to fill out T.

Initialization
for i = 0 to 255 do
S[i] = i;
T[i] = K[i mod keylen];

Further, we use T to produce the initial permutation of S. This involves starting with S[0] and going through to S[255], and, for each S[i], swapping S[i] with another byte in S according to a scheme dictated by T[i]:

Initial Permutation of S
j = 0;
for i = 0 to 255 do
j = (j + S[i] + T[i]) mod 256;
Swap (S[i], S[j]);

Because the only operation on S is a swap, the only effect is a permutation. S still contains all the numbers from 0 through 255.

Stream Generation: Once the S vector is initialized, the input key is no longer used. Stream generation involves starting with S[0] and going through to S[255], and, for each S[i], swapping S[i] with another byte in S according to a scheme dictated by the current configuration of S. After S[255] is reached, the process continues, starting over again at S[0]:

Stream Generation
i, j = 0;
while (true)
i = (i + 1) mod 256;
j = (j + S[i]) mod 256;
Swap (S[i], S[j]);
t = (S[i] + S[j]) mod 256;
k = S[t];

To encrypt, XOR the value of k with plaintext. For decryption same procedure is followed as the encryption but in reverse order, this is done by XORing the of value k with ciphertext.

c) Vigenère cipher: Vigenère[3,10] cipher is a kind of polyalphabetic ciphers which consist of a series of different Cesar ciphers for encryption. It consists of the alphabet written out 26 times in different rows, each alphabet shifted cyclically to the left compared to the previous alphabet, corresponding to the 26 possible Caesar ciphers shown in figure 6. In this table, the first row consists of n different elements and the remaining table has n-1 similar rows each proceeding one formed by left cycle shifting of elements of previous row. Decryption can be done by looking up the ciphered element in the row corresponding to the key element, then the column will represent the decrypted output, i.e. the original letter. Figure 5 shows the block diagram of Vigenère cipher. At different points in the encryption process, the cipher uses a different alphabet from one of the rows. The alphabet used at each point depends on a repeating keyword.

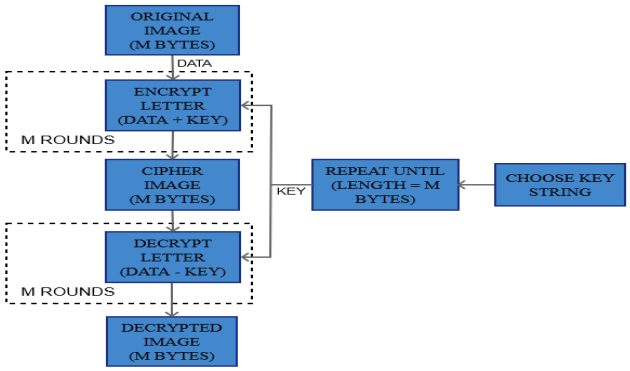

Figure 5. Block diagram of Vigenère cipher

For example, suppose that the plaintext to be encrypted is:

ATTACKATDAWN

The person sending the message chooses a keyword and repeats it until it matches the length of the plaintext, for example, the keyword "LEMON":

LEMONLEMONLE

Each row starts with a key letter. The remainder of the row holds the letters A to Z (in shifted order). Although there are 26 key rows shown, you will only use as many keys (different alphabets) as there are unique letters in the key string, here just 5 keys, {L, E, M, O, N}. For successive letters of the message, we are going to take successive letters of the key string, and encipher each message letter using its corresponding key row. Choose the next letter of the key, go along that row to find the column heading that matches the message character; the letter at the intersection of [key-row, msg-col] is the enciphered letter.

For example, the first letter of the plaintext, A, is paired with L, the first letter of the key. So use row L and column A of the Vigenère square, namely L. Similarly, for the second letter of the plaintext, the second letter of the key is used; the letter at row E and column T is X. The rest of the plaintext is enciphered in a similar fashion:

Plaintext: ATTACKATDAWN
Key: LEMONLEMONLE
Ciphertext: LXFOPVEFRNHR

Decryption is performed by going to the row in the table corresponding to the key, finding the position of the ciphertext letter in this row, and then using the column's label as the plaintext. For example, in row L (from LEMON), the ciphertext L appears in column A, which is the first plaintext letter. Next we go to row E (from LEMON), locate the ciphertext X which is found in column T, thus T is the second plaintext letter.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Figure 6. Vigenere Square

To recover the original data same procedure as the encryption process has to be followed.

3.1.2. Steganography Techniques Used:

a) LSB Substitution: In LSB substitution[1,11], some pixels of steganographic cover or cover image are selected. These pixels are selected randomly by using a particular seed which is known to both sender and receiver so as to generate same pixels during encryption and decryption[13]. The value of these pixels is converted into binary and the message bits are embedded in the place of Least Significant Bit. The number of pixels to be selected depends on the number of message bits to be embedded. The value of message bit is either 0 or 1 thus the change in pixel value is not very significant.

For the decryption procedure, the same pixels of the output image are selected by using the same seed used during encryption. After this, the LSB of these pixels are extracted which forms the data bits.

b) Distortion Substitution: This technique[11,9] is quite similar to LSB substitution due to the fact that it also selects some pixels of the cover image but instead of embedding data bit into least significant bit of the pixel, it changes the value of the selected pixel by adding a small value i.e. delta if the corresponding message bit is 1 and is left the same for the bit being 0.
For the process of decryption, we need the original image as well as the output image. The same pixels of both the images are selected using the seed used by the sender for encryption and the difference of these pixels are calculated which provides the data matrix.

3.1.3. Huffman Coding Technique

The procedure of Huffman coding[2] is shown in figure7 given below. In this method, a sequence of data symbols is coded with the minimum number of bits necessary. In Huffman coding, shorter codes are assigned to symbols that occur more frequently and longer codes to those that occur less frequently. To do this, firstly the probabilities of all the data symbols are calculated by the frequency of their occurrence in the message. After that, all the probabilities are arranged in decreasing order with their symbols. Then the least two probabilities are combined and the '0' is assigned to the upper probability and '1' is assigned to the lower probability. Then the probabilities are again lined up in decreasing order. This is repeated until there are only two probabilities left. To write the code, the path from the end is followed up to the symbol for which the code is required. Considering the following example, the code for the symbol 'a' is written '00' because the path that leads to it from the end consists of '0' followed by a '0'.

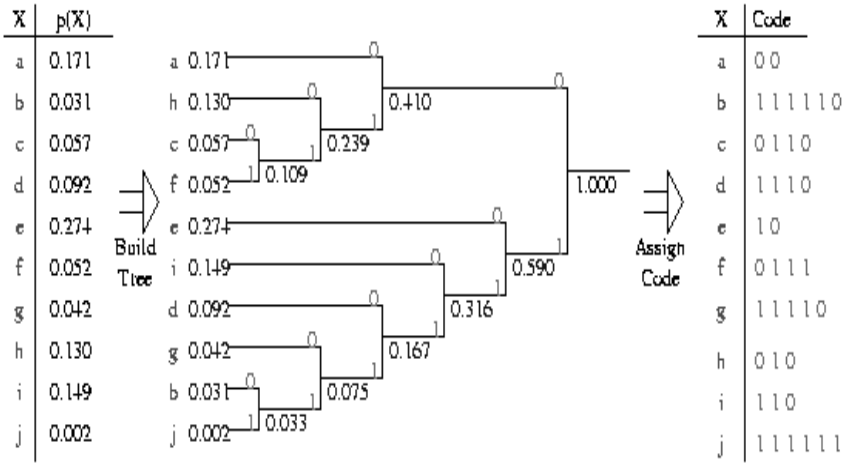

Figure 7: Example of Huffman Coding

3.2. Receiver Side Procedure

The image is received and the embedded data is extracted by reversing the steganography method that is used during encryption. This data is then applied to Huffman decoder and the ciphertext is obtained. The ciphertext is converted back into plaintext by using decryption key.

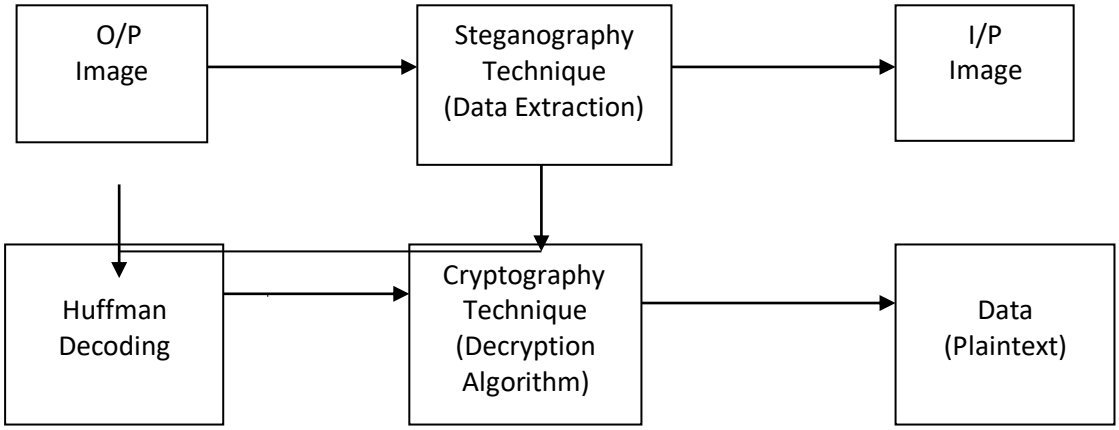

Figure 8: Mechanism for receiver

4. SIMULATION SETUP PARAMETERS

4.1. Performance Metric

The following parameters are used to comprise performance metric for comparison of results:

- **Peak Signal to Noise Ratio (PSNR) and Mean Square Error (MSE):** These two parameters represent the error introduced in the final image with respect to the original image due to embedded bits.

$$MSE = \frac{1}{XY}\sum_{i=0}^{X-1}\sum_{j=0}^{Y-1}[I(i,j) - K(i.j)]^2$$

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

Where, f (i, j) is the pixel value of original image.
g (i, j) is the pixel value of final embedded image.
X and Y are the dimensions of the image.
MAX is the maximum value of pixel in the image.

- **No. of Data Bits:** It is a comparison of data bits before and after applying huffman coding because it compresses the data.

- **Picture Quality:** The picture quality of the image is determined after embedding the data bits to notice any visible assessment.

- **Time Complexity:** The total time taken for encryption and decryption at sender and receiver encompasses time complexity.

4.2. Simulation Parameters

The following table shows the various set up parameters:

Table 2: Simulation Setup Parameters

Image Pixels (NxN)	512,256,128,64
Image Type	PNG
Encryption Algorithms	RC4,DES,Vigenere Square
Steganography Algorithms	LSB and Distortion Substitution
Seeds Used	100,102,103,104
RC4	String = 78,58,28 and 14 characters
DES	64 Bits
Simulation Tool	MATLAB 2016a

5. SIMULATION RESULTS

5.1. PSNR and MSE

The PSNR and MSE values are calculated by using the following equations:

$$MSE = \frac{1}{XY}\sum_{i=0}^{X-1}\sum_{j=0}^{Y-1}[I(i,j) - K(i.j)]^2$$

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

Where, f (i, j) is the pixel value of original image.
g (i, j) is the pixel value of final embedded image.
X and Y are the dimensions of the image.
MAX is the maximum value of pixel in the image.

- In case of 512x512 image, the maximum value is observed when Vigenere Square and Distortion Substitution is applied with Huffman coding and the minimum value is when RC4 and LSB Substitution is applied with Huffman Coding and the difference between them is 1.751.

- Without applying Huffman coding, the maximum value is with Vigenere square and Distortion substitution but the minimum value is when DES and LSB substitution applied and the difference between them is 1.885.

- The result shows that PSNR value increases when Huffman coding is applied. The average increase is from 1.0 to 1.7% (recommended figures 9(a),(b),(c),(d))

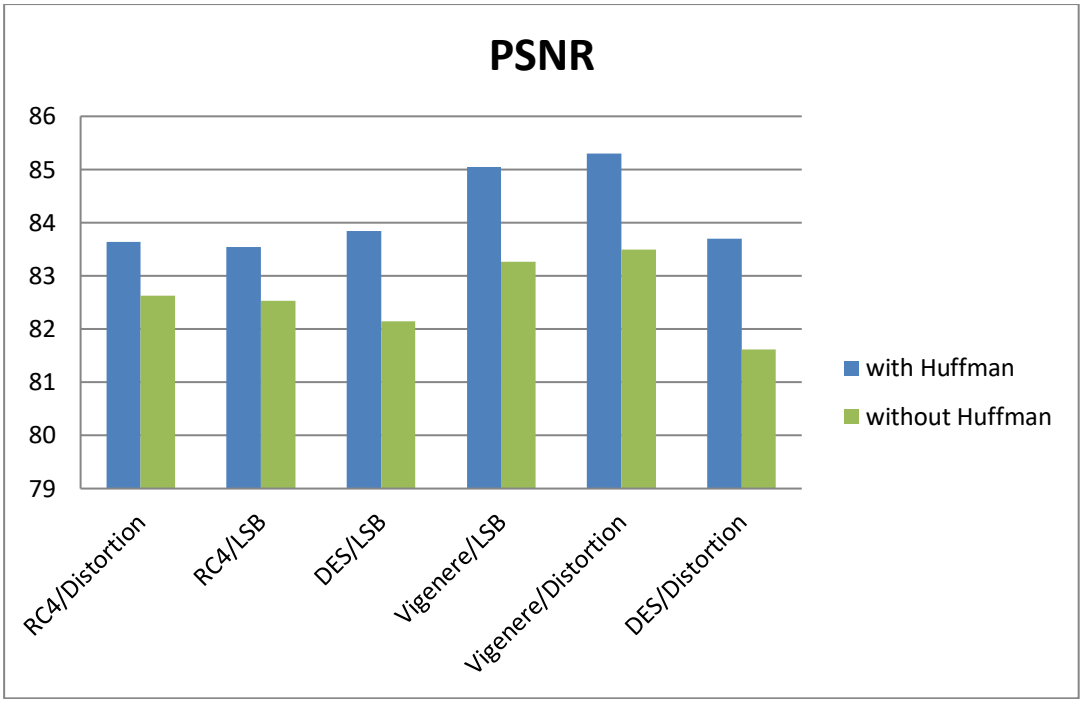

Figure 9(a) : Comparison of PSNR For 512x512 Image

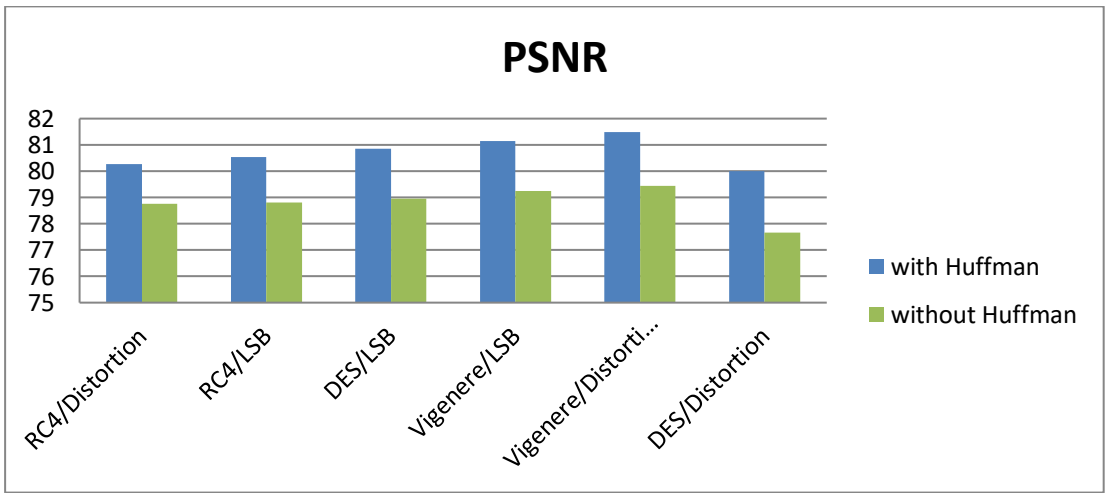

Figure 9(b): Comparison of PSNR for 256x256 Image

- The value of PSNR keeps on decreasing as the size of image decreases. The following graphs shows the change in PSNR for different size images:

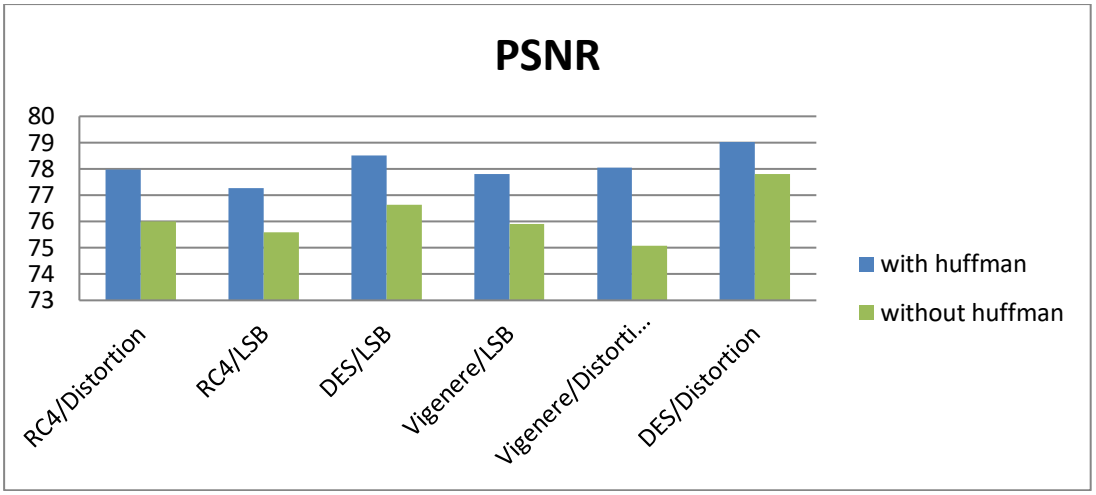

Figure 9(c) : Comparison of PSNR for 128x128 Image

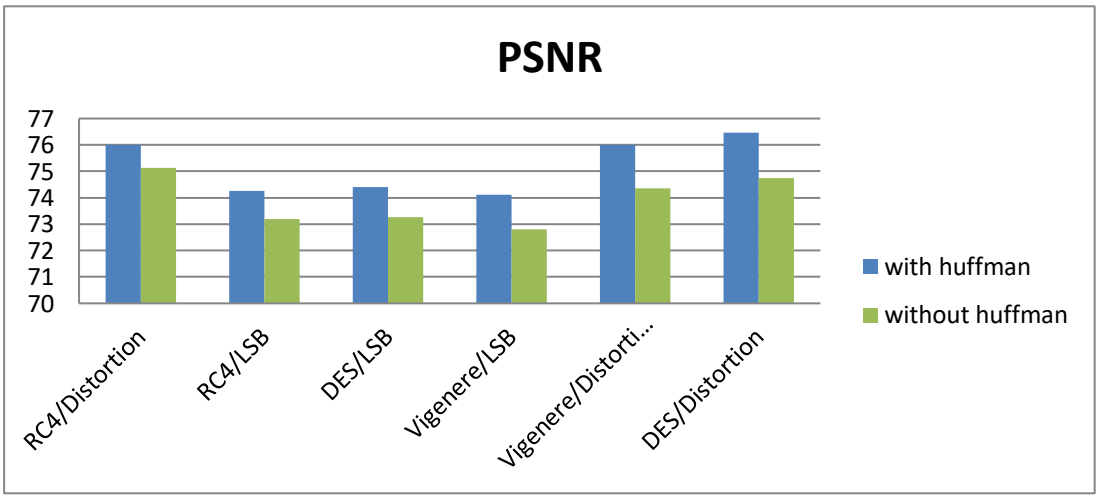

Figure 9(d) : Comparison of PSNR for 64x64 Image

- The values of MSE decreases when Huffman coding is applied which are shown in the following graphs for different image sizes. The results shown in figure 10(a),(b),(c),(d).

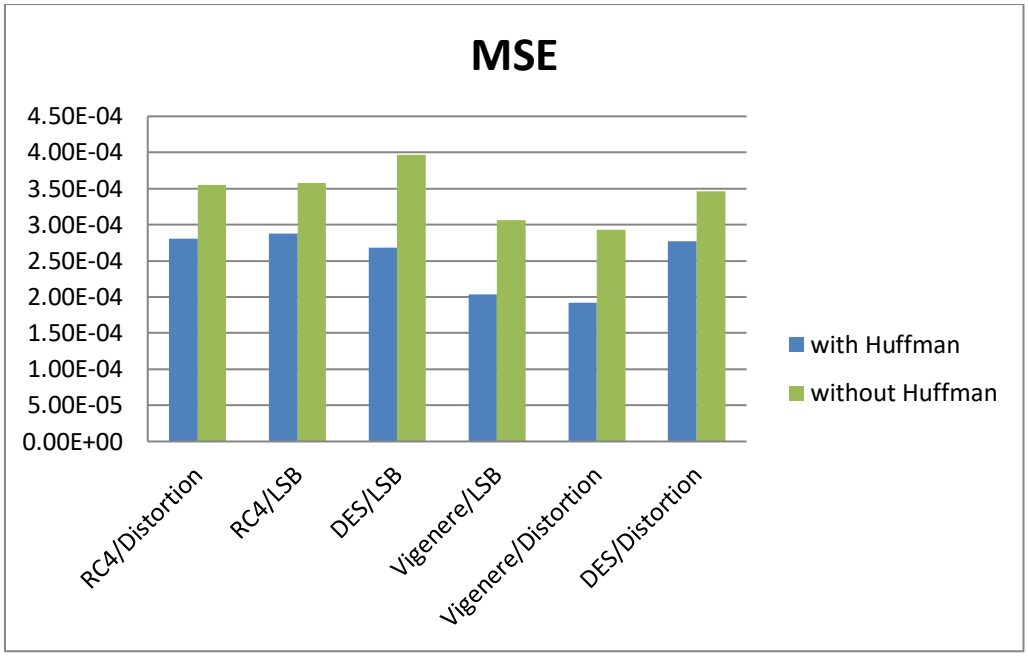

Figure 10(a): Comparison of MSE for 512x512 image

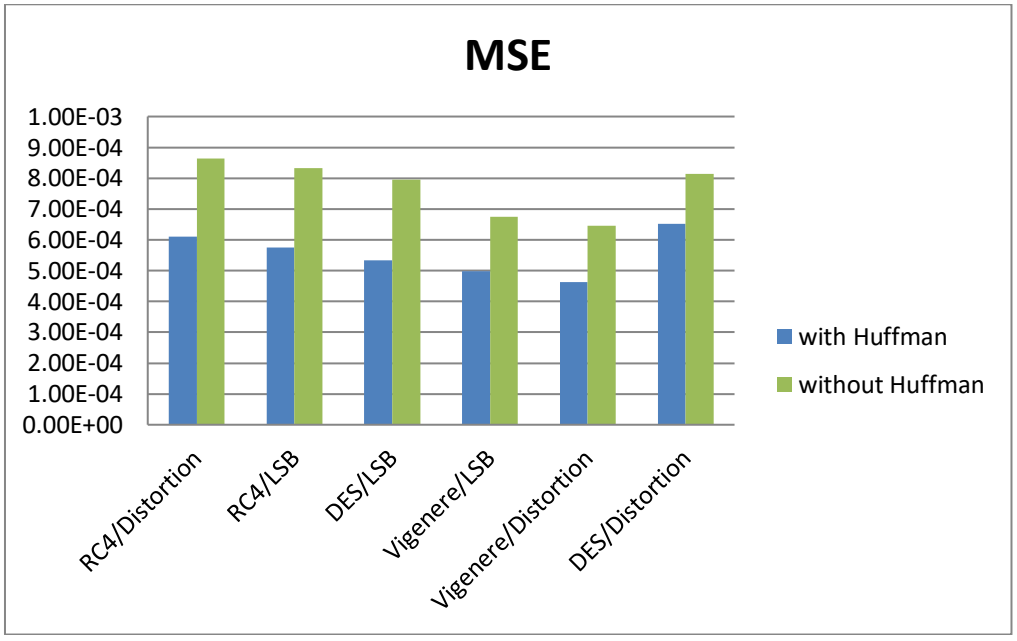

Figure 10(b): Comparison of MSE for 256x256 Image

- The value of MSE increases when the image size decreases because the number of data bits embedded in the image is less.

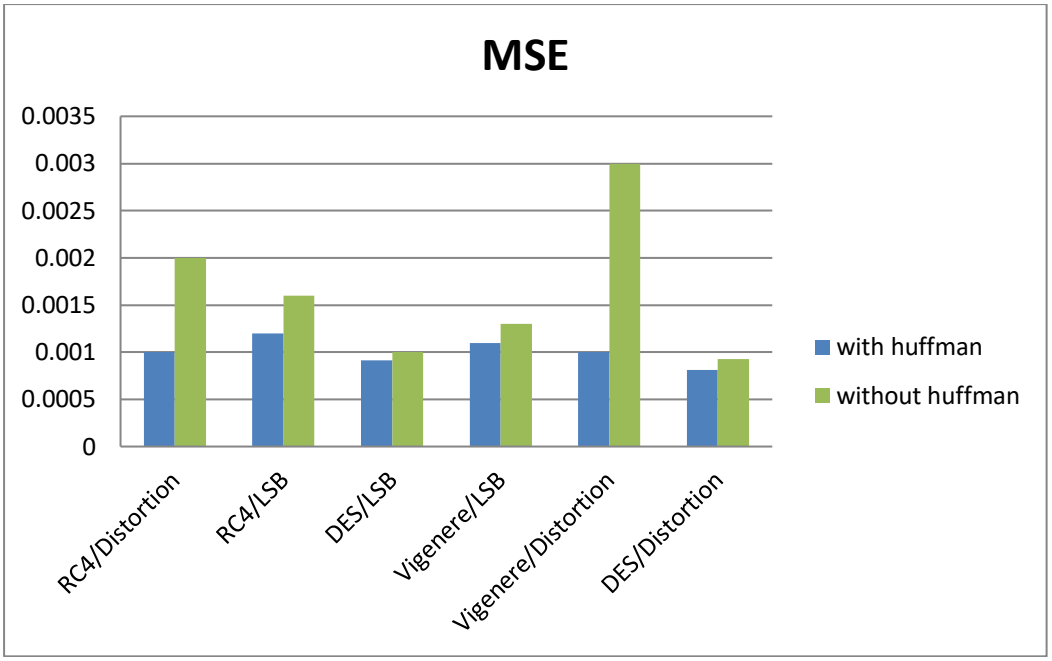

Figure 10(c) : Comparison of MSE for 128x128 Image

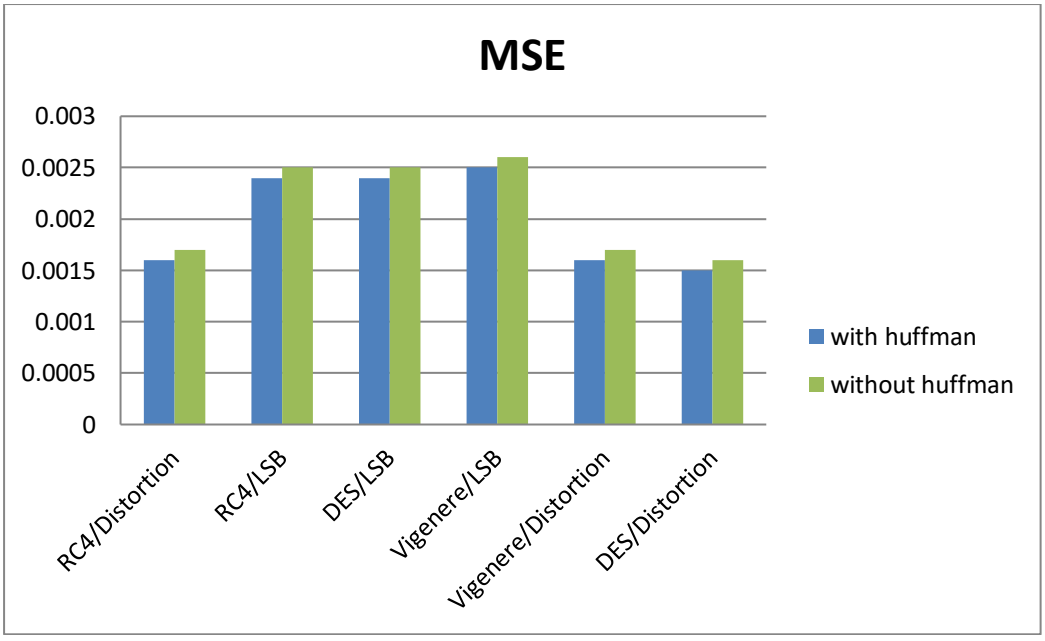

Figure 10(d): Comparison of MSE for 64x64 Image

5.2. Data Compression

- The use of Huffman Coding compresses the data bits from 52% to 65%.

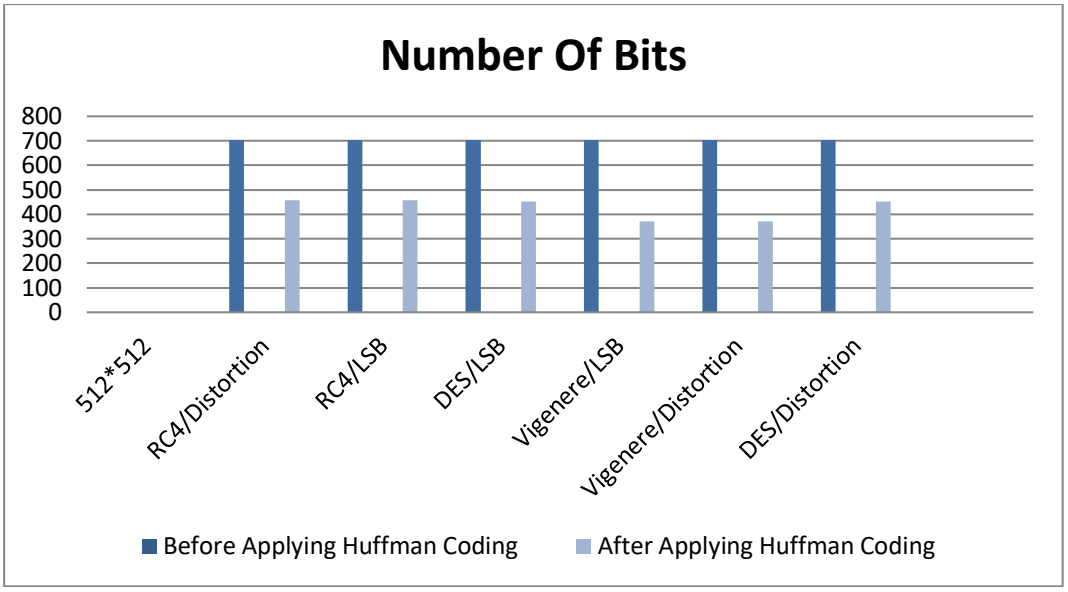

Figure 11(a): Comparison of Data Bits for 512x512 Image

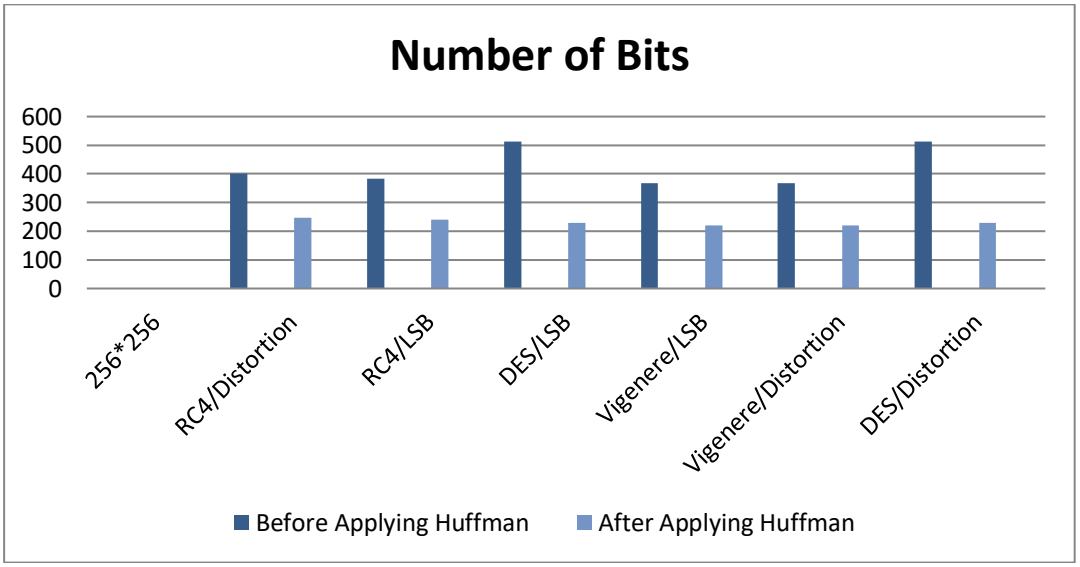

Figure 11(b): Comparison of Data Bits for 256x256 Image

- It is evident from results that the maximum number of bits decreases when RC4 is applied with LSB or Distortion Substitution and the minimum decrease is when Vigenere Square encryption is applied with LSB or Distortion Substitution.

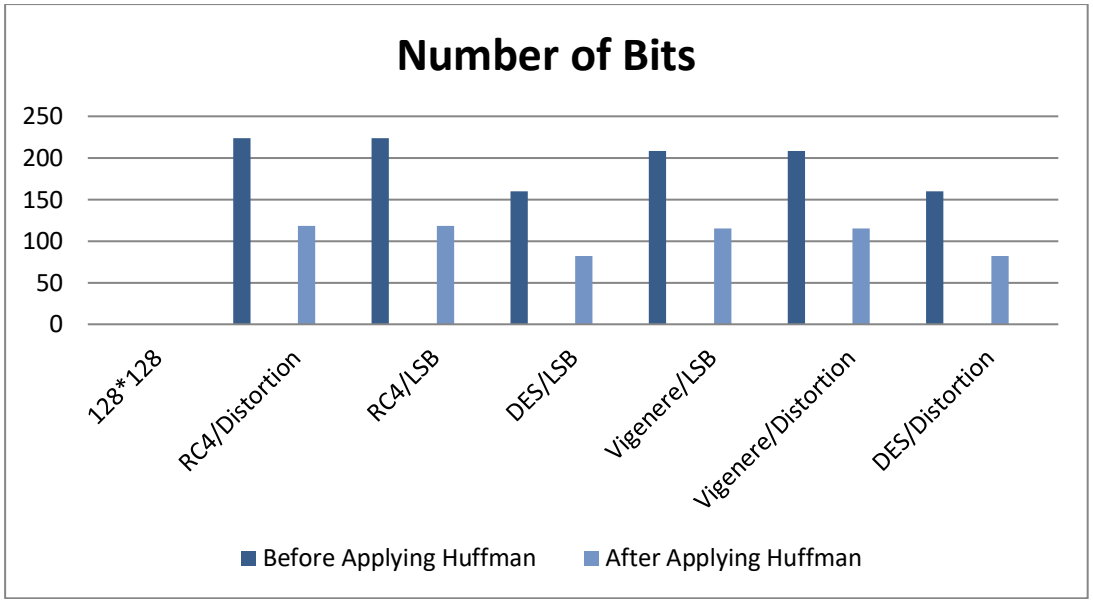

Figure 11(c): Comparison of Data Bits for 128x128 Image

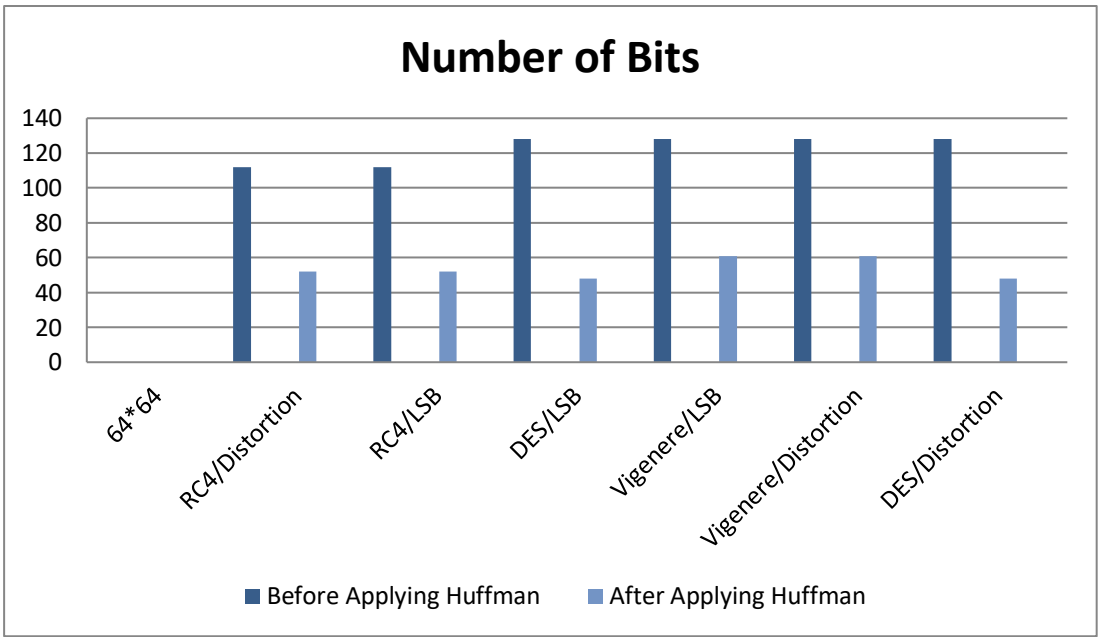

Figure 11(d): Comparison of Data Bits for 64x64 Image

5.3. Picture Quality

There are not any significant changes in the picture quality when any of the techniques applied. The following pictures show the original image and the final image after the bits are embedded:

Figure 12(a) : Original 512x512 Image

Figure 12(d) : Image after applying DES,LSB and Huffman

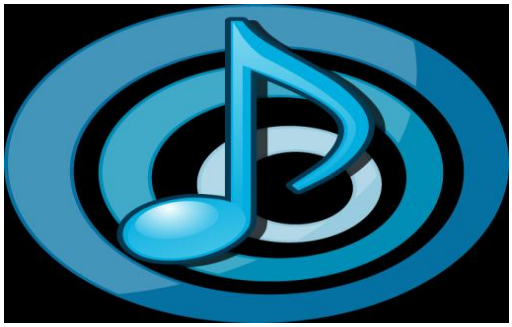

Figure 12(b) : Image after applying RC4, Distortion and Huffman

Figure 12(e) : Image after applying Vigenere Square, LSB and Huffman

Figure 12(c) : Image after applying RC4,LSB and Huffman

Figure 12(f) : Image after applying Vigenere Square,Distortion and Huffman

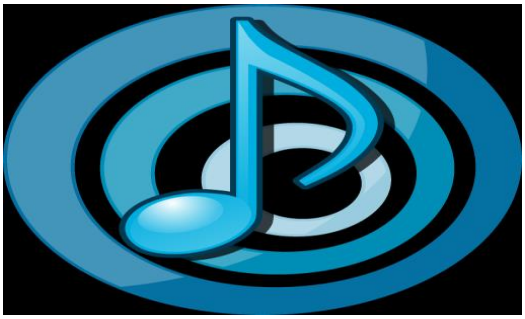

Figure 12(g) : Image after applying DES, Distortion and Huffman

- LSB and Distortion substitution changes the pixel values into a very small extent .therefore; the picture quality is almost same.

Figure 13(a) : Original 256x256

Figure 13(b) : After applying RC4,LSB Image and Huffman

Figure 13(c) : After applying DES,LSB and Huffman

Figure 13(d) : After applying Vigenere, LSB and Huffman

Figure 13(e) : After applying Vigenere, Distortion and Huffman

Figure 13(f) : After applying DES, Distortion and Huffman

- Smaller the size of the image, less number of data bits are embedded so as not to distort the image and make it prone to steganalysis.

Figure 14(a) : Original 128x128 Image

Figure 14(b) : After applying RC4,LSB and Huffman

Figure 14(c) : After applying DES,LSB and Huffman

Figure 14(d) : After applying Vigenere, LSB and Huffman

Figure 14(e) : After applying Vigenere,

Figure 14(f) : After applying DES,

Distortion and Huffman

Figure 15(a) : Original 64x64 Image

Distortion and Huffman

Figure 15(b) : After applying RC4, LSB and Huffman

Figure 15(c) : After applying DES, LSB and Huffman

Figure 15(e) : After applying Vigenere, LSB and Huffman

Figure 15(f) : After applying Vigenere, Distortion and Huffman

Figure 15(g) : After applying DES, Distortion and Huffman

5.4. Time Complexity

- The time taken by the various techniques is slightly higher when Huffman coding is used but security mechanism is greatly improved and more data bits can be embedded in the image.

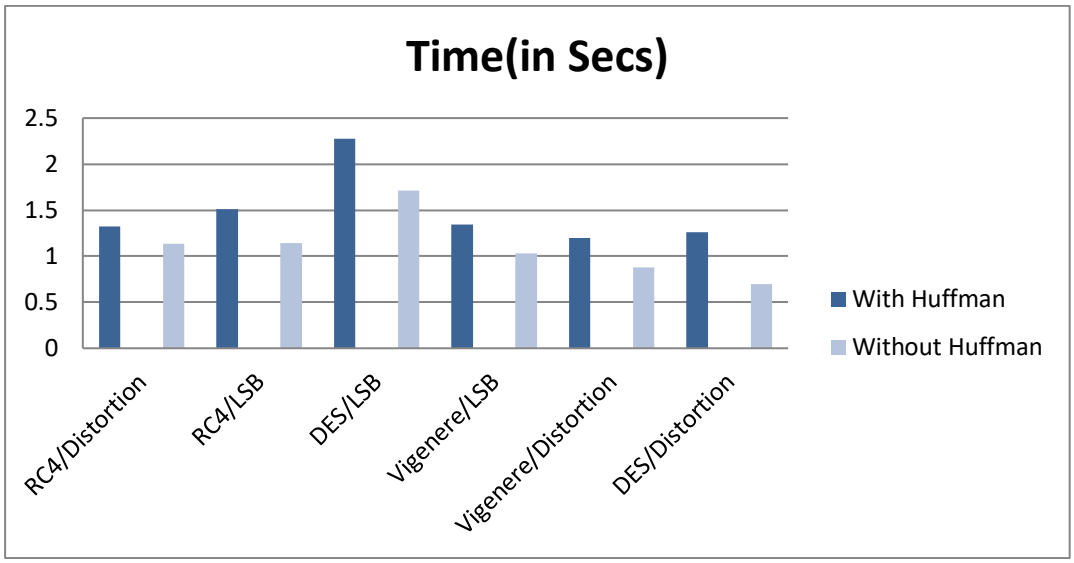

Figure 16(a): Time Complexity for 512x512 Image

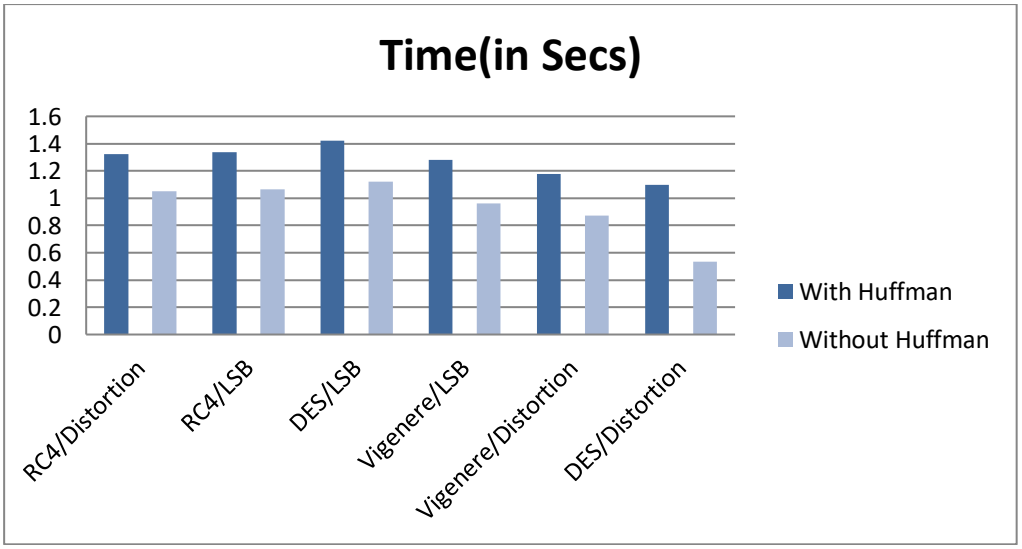

Figure 16(b): Time Complexity for 256x256 Image

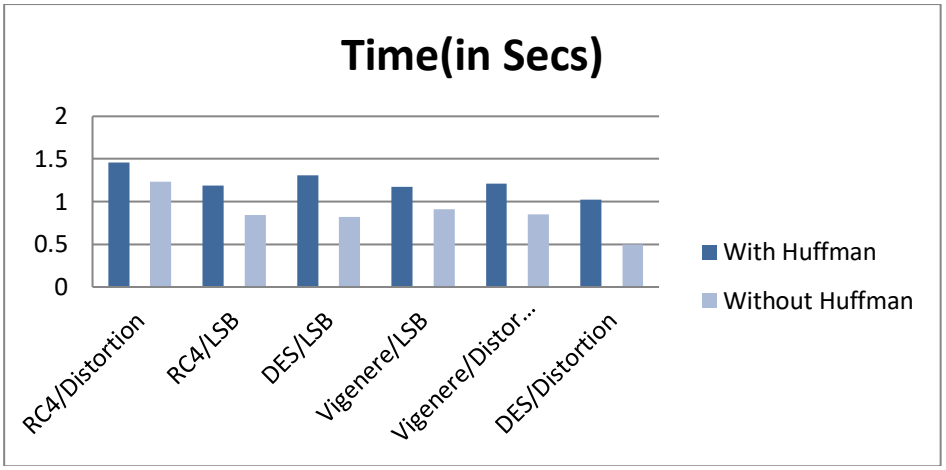

Figure 16(c): Time Complexity for 128x128 Image

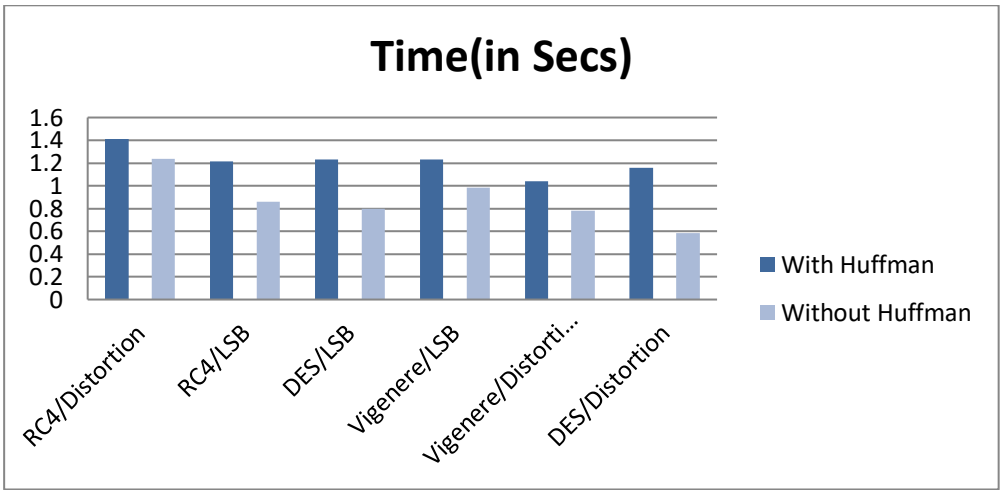

Figure 16(d): Time Complexity for 64x64 Image

6. CONCLUSION AND FUTURE SCOPE

The mechanism proposed in this paper tries to compress the data which is being sent through the cover image by using Huffman Coding. It also provides enhanced security by using the combination of Cryptography, Steganography and Huffman encoding which have not been used earlier.

- The time complexity marginally increases with the use of Huffman coding but it increases security upto a great extent because it compression cipher text takes place and the data bits changes all together.
- As the data bits embedded to be decreases, the distortion in the steganographic cover will be less and it will be secure from steganalysis.
- Even if the adversary is able to find the data bits from cover image, still it will be very difficult to decompress and decipher them thus increasing the brute force search time.

REFERENCES

[1] N. F. Johnson and S. Jajodia, **(1998)** "Steganalysis of Image Created Using Current Steganography Software", Workshop of Information Hiding Proceedings, Portland Oregon, USA, vol. 1525, 15-17, Lecture Notes in Computer Science, Springer-Verlag.

[2] Pankaj Kumar and Ankur Kumar Varshney**, (2012) "**Double Huffman Coding", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 8**.**

[3] W. Stallings, **(2003)** "Cryptography and Network Security: principles and practices", Pearsons education, first Indian reprint.

[4] "Applied Cryptography", (**1996)** 2nd Edition, John Wiley & Sons, Inc., Bruce Schneier, New York.

[5] S. Fluthrer, I. Mantin, A. Shamir, (**2001)** "Weaknesses in the Key Scheduling Algorithm of RC4",published in SAC2001 (S. Vaudenay, A. Youssef,eds.), col. 2259 of LNCS, pp. 1-24, springer-Verlag.

[6] "Advanced Encryption Standard Call", **(2001)** the symmetric block cipher block cipher ratified as a standard by National Institute of standards and Technology of United States NIST,FIPSP 197.

[7] Matsui, M., **(1994)** "The first experimental cryptanalysis of the Data Encryption Standard" , Advances in Cryptology - CRYPTO '94:1–11. doi: 10.1007/3-540-48658-5_1.

[8] "National Bureau of Standards - Data Encryption Standard", **(1977)** published in Federal Information Processing Standards Publication, FIP PUB- 46.

[9] W. Bender, D. Gruhl and N. Morimoto, **(1996)** "Techniques for data hiding", IBM Systems Journal , vol. 35, no. 3/4.

[10] Kester, Q. A. **(2012)** "A hybrid cryptosystem based on Vigenère cipher with varying key" published in International Journal of Advanced Research in Computer Engineering & Technology (IJARCET),Volume 1, Issue 10, ISSN: 2278 – 1323, pp 108-113.

[11] S. A. Moller, Pitzmann and I. Stirand, **(1996)** "Computer Based Steganography: How It Works and Why Therefore Any Restrictions on Cryptography Are Nonsense, At Best", in Information Hiding: First International Workshop, Proceedings, vol. 1174 of Lecture Notes in Computer Science, Springer.

[12] D. Gruhl, A. Lu and W. Bender, **(1996)** "Echo Hiding in Information Hiding", First International Workshop, Proceedings, vol. 1174 of Lecture Notes in Computer Science, Springer.

[13] S. H. Low, N. F. Maxemchuk and A. M. Lapone, **(1998)** "Document Identification for Copyright Protection Using Centroid Detection", IEEE Transactions on Communications, vol. 46, no. 3.

[14] G. B. Rhodas, **(1998)** "Method and Apparatus Responsive to a Code Signal Conveyed Through a Graphic Image", U.S. Patent 5,710,834.

[15] Aamer Nadeem, Dr. M. Younus Javed, (**2005)**"A Performance Comparison of Data Encryption Algorithms" published in the proceedings of IEEE conference, pp 84-89.

[16] S. Nararayana and G. Prasad, **(2010)** "Two New Approaches For Secured Image Steganography Using Cryptographic Technique and Type Conversions", Signal and Image Processing: An International Journal (SIPIJ), vol. 1, no. 2, December.

[17] S. Gupta, A. Goyal and B. Bhushan, **(2012)** "Information Hiding Using Least Significant Steganography and Cryptography", I. J Modern Education and Computer Science, vol. 6, IJMECS.

[18] S. Gupta, B. Bhushan, S. Singhania and J. Gulani **(2013)** "A Hybrid approach for ensuring security in data communication", Accepted for publication in CCSIT 2013.

[19] H. Sheisi, J. Mesgarian and M. Rahmani, **(2012)** "Steganography: DCT Coefficient Replacement Method and Compare with Jsteg Algorithm", International Journal of Computer and Electrical Engineering, vol. 4, no. 4.

[20] Pardeep, Pushpendra Kumar Pateriya**, (2012) "**PC1-RC4 and PC2-RC4 Algorithms: Pragmatic Enrichment Algorithms to Enhance RC4 Stream Cipher Algorithm", International Journal of Computer Science and Network (IJCSN) Volume 1, Issue 3**.**

[21] Nimmi Gupta**, (2012) "** Implementation of Optimized DES Encryption Algorithm upto 4 Round", International Journal of Computer Technology and Electronics Engineering (IJCTEE) Volume 2 , Issue 1 **.**

[22] Surbhi Singhania, Shailender Gupta, Bharat Bhushan and Ajay Nain, **(2013)** "A Novel Crypt-Stego Technique for Information Security in Communication Networks"**,** International Journal of Signal Processing, Image Processing and Pattern Recognition Vol. 6, No. 2**.**

AUTHORS

Manju Kumari received Master of Technology in Electronics & Communication Engineering from Deenbandhu Chotu Ram University of science and Technology,Murthal,India,in 2011 and persuing PhD.in Electronics & Communication Engineering from YMCA University of Science and Technology,Faridabad,India. She is currently working as assistant professor since 2012 in YMCA University of Science and Technology,Faridabad,India. Her research interest include Image Processing, Network Security and Wireless Sensor Networks.

Vipin Pawar received Master of Technology in Electronics & Communication Engineering from Deenbandhu Chotu Ram University of science and Technology,Murthal,India,in 2012. He completed his Bachelor in technology degree from Bhagwan Mahavir Institute of Science and Technology,Sonipat,India in 2009. His research interest include Image Processing, Network Security and Nano-Technology.

Pawan Kumar Received M.Sc in Mathematics from CCS University, Meerut (UP), India . He completed his bachelor in B.Sc (H) Mathematics from University of Delhi and Persuing PhD.in Study of fixed point theorem from **Dr. A.P.J. Abdul Kalam Technical University Uttar Pradesh, Lucknow** ,India. He is currently working as Assistant Professor since 2012 in Maitreyi college , University of Delhi, India.