

**Abstract**

Visual cryptography is a method for protecting image-based secrets that have a computation-free decoding process. In this technique, number of shares are generated from one image. The shares are sent to the receiver and the receiver generates the original image by stacking all the share images. The method is fair but wasting the bandwidth of the network. The technique of generating shares is different in different types of images like binary, gray and color images. In this paper, an algorithm has been proposed to convert image in encrypted form and decrypt the image into original form. The key of encryption and decryption has been generated automatically using a real number. This number can be predefined or can be sent by any means to the receiver. The proposed algorithm can be applied to any type of image i.e. binary, gray scale and color image. The distribution of pixels in encrypted image is quite similar to a single share. Finally, it has been proved that the proposed approach is much simpler, secure and consumes less bandwidth of network.

**Introduction**

In 1949, Claude Shannon introduced the idea of substitution-permutation networks. This idea is the basis of modern block ciphers. The block ciphers are composed of two basic operations: substitutions and permutations. The same approach has not been used in visual cryptography. Naor & Shamir have applied the concept in images in their paper "Visual Cryptography". They extend their new scheme to secret sharing problem. In this paper, they have planted the seed of the visual cryptography and visual secret sharing. The authors extended visual secret sharing into a visual variant of the  $k$  out of  $n$  secret sharing. They have presented a problem in which a dealer provides a transparency to each one of the  $n$  users any  $k$  of them can see the image by stacking their transparencies, but any  $k-1$  of them gains no information about it. The basic approach was to split the image into 2 shares that generated from the original secret image and by stacking together the secret is revealed. This approach was restricted in binary images which is insufficient in real time applications. Chang-ChouLin, Wen-Hsiang Tsai proposed Visual Cryptography for gray level images by dithering techniques. Instead of using gray sub pixels directly to constructed shares, a dithering technique is used to convert gray level images into approximate binary images. But all of the methods suffer from a severe limitation, which hinders the objectives of Visual Cryptography. The limitation lies in the fact that all shares are inherently random patterns carrying no visual information, raising the suspicion of data encryption. Mizuho Nakajima and Yasushi Yamaguchi[10] proposed extended visual cryptography for natural images constructs meaningful binary images as shares. (Refer Figure 1)

In this paper, an algorithm has been proposed in visual cryptography without using share concept. The input image has been encrypted by image element and key value. The distribution of image gray values is in such a way that the encrypted image is treated as single share. The encrypted image is decrypted by same proposed algorithm. The key is generated automatically by using a number. The value of the number is taken from 0 to 1. This value can be pre-agreement between the sender and receiver or it can be sent by any secret means to the receiver. The result of this approach is much more secure and less bandwidth consuming. This type of

Mr. Satyendra Nath Mandal  
Dept. of I.T, Kalyani Govt. Engg. College, Kalyani, Nadia(W.B)

Mr. Subhankar Dutta  
Dept. of CSE, Kalyani Govt. Engg College, Kalyani, Nadia(W.B)

Mr. Ritam Sarkar  
Dept. of CSE, Kalyani Govt. Engg College, Kalyani, Nadia(W.B)

efforts has not been tested earlier which is the reason of making this paper.

### Existing Technique of Visual Cryptography

#### 2-out of -2 Visual Cryptography Scheme

In this concept one white or black pixel will divide into two sub pixel. One way combination of the pixel division is shown in figure 2. It is mentioned that the shares 1 and 2 are stacked together and they get the result in the form of complete black or gray (it's partially white and black but visualizes as gray). Because of this when we stacked the shares the white in original secret image become gray in the stacked result. (Refer Figure II)

#### Visual Cryptography for Gray Level Images

Previous efforts in visual cryptography were restricted to binary images which were insufficient in real time applications. Chang-ChouLin, Wen-Hsiang Tsai proposed visual cryptography for gray level images by dithering techniques. Instead of using gray sub pixels directly to constructed shares, a dithering technique is used to convert gray level images into approximate binary images. Then existing visual cryptography schemes for binary images are applied to accomplish the work of creating shares. The effect of this scheme is still satisfactory in the aspect of increase in relative size and decoded image quality, even when the number of gray levels in the original image still reaches 256.

#### Extended Visual Cryptography for Natural Images

All of the Visual Cryptography methods suffer from a severe limitation, which hinders the objectives of Visual Cryptography. The limitation lies in the fact that all shares are inherently random patterns carrying no visual information, raising the suspicion of data encryption. Mizuho Nakajima and Yasushi Yamaguchi[10] proposed extended visual cryptography for natural images constructing meaningful binary images as shares. This will restrain the cryptanalysts to suspect secrets from an individual share. The previous research basically handled only binary images and they established the extended Visual Cryptography scheme suitable for natural images is shown in Figure III.

#### Visual Cryptography for Color Images

The research in visual cryptography leads to the degradation in the quality of the decoded binary images, which makes it unsuitable for protection of color image. F. Liu, C. K. Wu X.J. Lin proposed a new approach on Visual Cryptography for colored im-

ages. They proposed three approaches which are as follows:

1. The first approach to realize color Visual Cryptography is to print the colors in the secret image on the shares directly similar to basic model. It uses larger pixel expansion which reduces the quality of the decoded color image.
2. The second approach converts a color image into black and white images on the three color channels (red, green, blue or equivalently cyan, magenta, yellow), respectively, and then apply the black and white visual cryptography to each of the color channels. This results in decrease of pixel expansion but reduces the quality of the image due to halftone process.
3. The third approach utilizes the binary representation of the color of a pixel and encrypts the secret image at the bit-level. This results in better quality but requires devices for decryption. (Refer Figure IV and V).

### Proposed Algorithm

#### Key Generation

Input: Original Image and a real number between 0 to 1

Output: Key

Method:

Step 1: To declare variables BinaryValue (an array initialized to zero), BinaryValueTemp (temporary container of bin value),  $x_N$ .

Step 2: To initialize a loop to put the values of BinaryValueTemp to  $x_N$  using the equation:

For index =2 to N //  $N = \text{ImageHeight} * \text{ImageWidth} * 8$

$x_N = 1 - 2 * \text{BinaryValueTemp} * \text{BinaryValueTemp}$

if ( $x_N > 0.0$ )

BinaryValue[index-1]=1

else

BinaryValueTemp= $x_N$

End if

Next index

Step 3: To Initialize an array to hold the key and assign the value to zero initially.

Step 4: To create the key array using a repetitive sum method using the formula

```

For i= 1 to n*m // ImageHeight*ImageWidth
    For j=1 to 8
        Key[i] = Key[i] +
        BinaryValue[i*j] * 2 ^ (j-1)
    Next j
Next i

```

### Algorithm for Encryption

Input: Original Image and Key

Output: Encrypted Image

Method:

Step1: To retrieve the key from Key generation step.

Step 2: Assign the value of the key array using the principal diagonal approach(i.e. first row first column, second row first column, third row first column, fourth row first column, first row second column....this creates the block wise approach) to a new array FinalKey.

Step 3: To retrieve the image length and breadth and use it as index values.

Step 4: To perform bitwise XOR to the browsed image with the FinalKey value.

Step 5: Print encoded image

### Algorithm for Decryption

Input: Encrypted image and the real number between 0 and 1 used in key generation

Output: Decrypted image

Method:

Step 1: Use the key generation step again to retrieve the Key array.

Step 2 : Assign the value of the key array using the principal diagonal approach(i.e. first row first column, second row first column, third row first column, fourth row first column, first row second column....this creates the block wise approach) to a new array FinalKey.(This is now same as the original key assigned in part 1).

Step 3: Retrieve the image length and breadth and use it as index values.

Step 4: To perform bitwise XOR to encoded image with the FinalKey value.

Step 5: Print decrypted image.

## Implementation

### Illustrate the Algorithm with Example

The algorithm deals with a new aspect of image encryption and decryption regarding key generation. Here is a brief description of it.

Step 1: Let us consider a RGB image (to be encrypted) having image length n and image width m.

Step 2: Let us take a one dimensional array having (n\*m\*8) number of elements (here 8 refer to the block size). We call this array BinaryValue as it only takes binary values 0 and 1. For the sake of the algorithm we initialize BinaryValue to zero. For the sake of simplicity we assume n=4 and m=4.here n\*m = 4\*4 = 16.

BinaryValue (initialized)

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Step 3: Depending on a temporary variable named BinaryValueTemp we assign arbitrary values of 0 and 1 at various positions of BinaryValue. This variable BinaryValueTemp holds the entire responsibility of generating the key. We do not need to send the entire key to the receiver. Instead we only need to send the value of BinaryValueTemp. The following diagram shows some arbitrary values of BinaryValue after the operation with BinaryValueTemp

BinaryValue (after the operation)

0	1	0	1	1	0	0	1	1	0	0	1	0	1	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Step 4: We use a repetitive sum method to generate the key array which is a one dimensional array having the dimension (n\*m). The values in the key array will range from 0 to 255.

Key

129	72	112	82	204	4	94	145	82	42	86	16	99	3	173	8
-----	----	-----	----	-----	---	----	-----	----	----	----	----	----	---	-----	---

Step 5: In the next step we generate a two dimensional array named FinalKey having n number of rows and m number of columns. Now we will map the values of the one dimensional key array to this two dimensional FinalKey array. Refer Figure VI.

The principal diagonal splits the FinalKey array into two sub parts. Left triangular matrix and right triangular matrix. (Refer Figure VII). In our algorithm the left triangular matrix will have higher precedence than the right subpart. For example we have assumed the block size to be 8. So the block will contain 4 rows and two columns instead of 2 rows and 4 columns.

Let us consider FinalKey (1,1) to be the root. So initially the principal diagonal will start from that position and Key (1) will be entered in that position. Next FinalKey(2,1) will be the left most element of the principal diagonal will be selected and key(2) will be assigned to that position. This will continue in case of FinalKey (3, 1) and FinalKey (4, 1). After the completion of the first column key (5) will be assigned to FinalKey (2, 1), key(6) will be assigned to FinalKey(2,2) and so on. The Figure VIII shows the one dimensional to two dimensional mapping of key values.

Step 6: Next we will use bitxor algorithm on the image and the FinalKey array. The encryption will be done serially i.e.

OriginalImage (1, 1) bitxor FinalKey (1, 1) = EncryptedImage (1, 1)..

OriginalImage (2, 1) bitxor FinalKey (2, 1) = EncryptedImage (2, 1)..

OriginalImage (1, 2) bitxor FinalKey (1, 2) = EncryptedImage (1, 2).. and so on...

Step 7: The decryption process will be similar to the encryption process. (Refer Figure IX)

Next we will use bitxor algorithm on the encrypted image and the FinalKey array. The encryption will be done serially i.e.

EncryptedImage (1, 1) bitxor FinalKey (1, 1) = DecryptedImage (1, 1)..

EncryptedImage (2, 1) bitxor FinalKey (2, 1) = DecryptedImage (2, 1)..

EncryptedImage (1, 2) bitxor FinalKey (1, 2) = DecryptedImage (1, 2).. and so on... (Refer Figure X)

Software package based on Proposed Algorithm

Front Page of the software

Refer Figure XI: Front page of the interface

Select the image to be encrypted

Refer Figure XII: Interface after browsing the original image

The original and encrypted image

Refer Figure XIII: Original and encrypted image after encryption

The original, encrypted and decrypted image

Refer Figure XIV: Original encrypted and decrypted after total process done

## Results

The algorithm has been tested on a different number of bench-mark or non-bench mark images are furnished in the Figure XV.

Results after the application of the algorithm on different images

## Conclusion and Future Work

On the basis of the observed experimental results, it can be said that proposed algorithm has an extremely superior level of security. Above all, the proposed algorithm in Visual Cryptography is simple, straightforward but intrinsically strong and compact approach to cryptography. In order to hide the secrecy, the numbers of shares have been increased, but this affects the resolution. Therefore, an optimum number of shares are required to hide the secrecy. In the proposed method, the network overhead has been decreased and the image has been converted into secured encrypted image. A comparison has to be made between the proposed algorithm with other existing algorithms in respect of security, bandwidth, time and space complexity.

## References

- Uttam Kr. Mondal, Satyendra Nath Mandal, J. Pal Choudhury, J.K.Mandal(2008). "A New Approach to Cryptography", *International Conference Systematics, Cybernetics & Informatics (ICSCI 2008)*, Page 1-12.
- C.E. Shannon (1949), "Communication Theory of Security System", *Bell, System Technical Journal*, vol 28, pp.656-715,1949.
- Nalini. N and G. Raghavendra Rao, "A New Encryption and Decryption Algorithm Combining the Features of Genetic Algorithms (GA) and Cryptography"
- H. Feistel (1973). "Cryptography and Computer Privacy", *Scientific American* vol. 228, no. 5, pp 15-23, 1973.
- Behrouz A.Forouzan (2007). "*Cryptography & Network Security*", Tata McGraw Hill, ISBN 13-978-0-07-066046-5.
- Chang-Chou Lin, Wen-Hsiang Tsai, *Visual cryptography for gray-level images by dithering techniques*, Pattern Recognition Letters, v.24 n.1-3.
- A. Tragha, F.Omary and A. Mouloudi (2005).

“Genetic Algorithm Inspired Cryptography”, *A.M.S.E. Association for the Advancement of Modeling & Simulation Techniques in Enterprises, Series D: Computer Science and Statistics, November 2005.*

- Naor, M., and Shamir, A. (1995). Visual cryptography, in "Advances in Cryptology Eurocrypt

'94", *A. De Santis, Ed., Lecture Notes in Computer Science, Vol. 950, pp. 1-12, Springer-Verlag, Berlin.*

- Nakajima, M. and Yamaguchi, Y. (2011), "Extended visual cryptography for natural images". *Journal of WSCG, Vol. 10, issue 2, Pg. 303-310.*



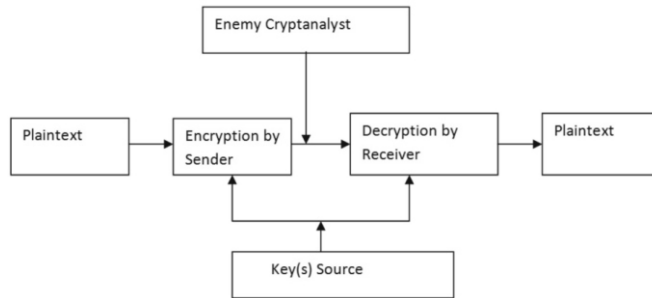


Figure I : Cryptosystem Proposed by Shannon

















Pixel	White		Black	
				
Prob.	50%	50%	50%	50%
Share 1				
Share 2				
Stack share 1 & 2				

Figure II: Basic Concept of 2-out-of-2 Visual Cryptography

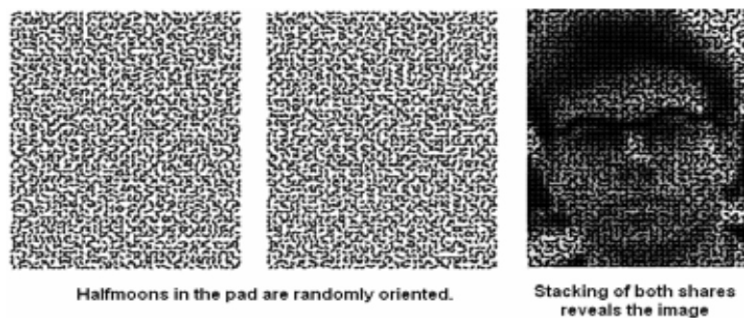


Figure III: Example of the Extended Visual Cryptography

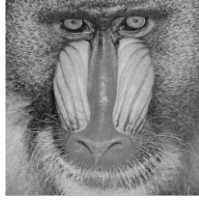


Figure IV : Original Secret Image Components

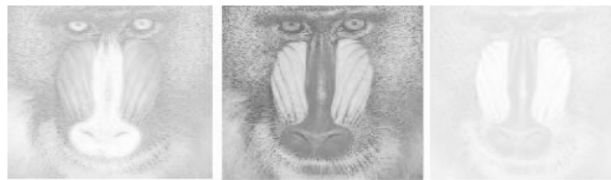


Figure V : Primitive Color (C,M,Y)

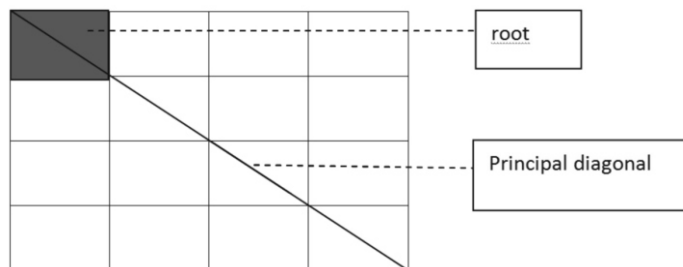
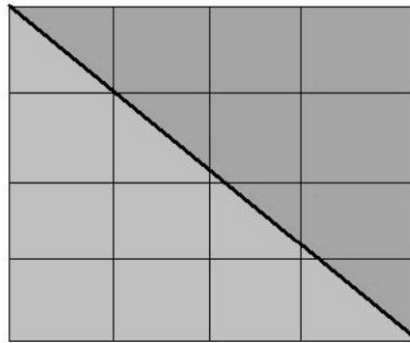


Figure VI: Principal Diagonal and the Root



Left triangular Matrix   
 Right triangular Matrix

Figure VII: Left and Right Triangulr Matrix w.r.t. the Principal diagonal

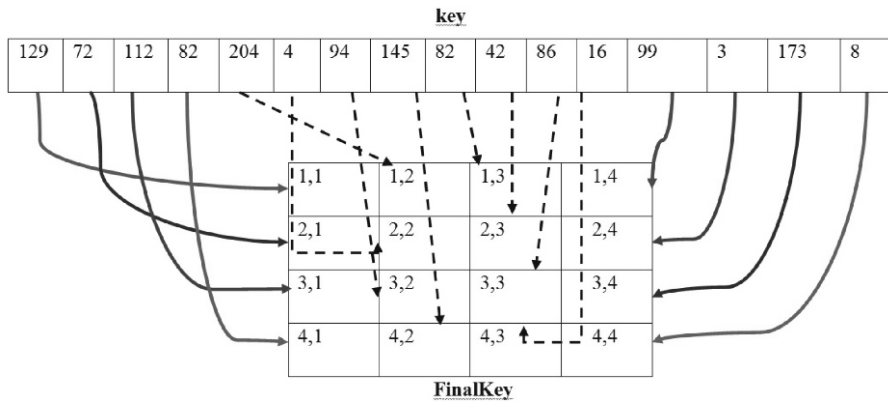


Figure VIII: One Dimensional to Two Dimensional Mapping of key Values



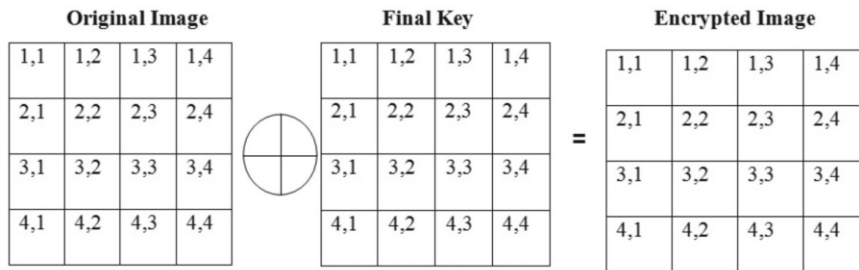


Figure IX: Bitxor Algorithm on the Image and the Final Key array

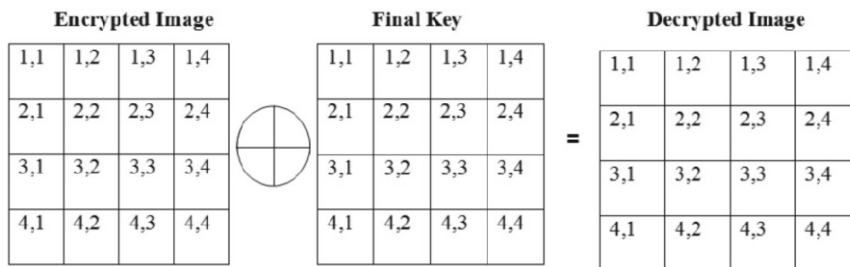


Figure X : Bitxor Algorithm on the Encrypted Image and the Final Key Array

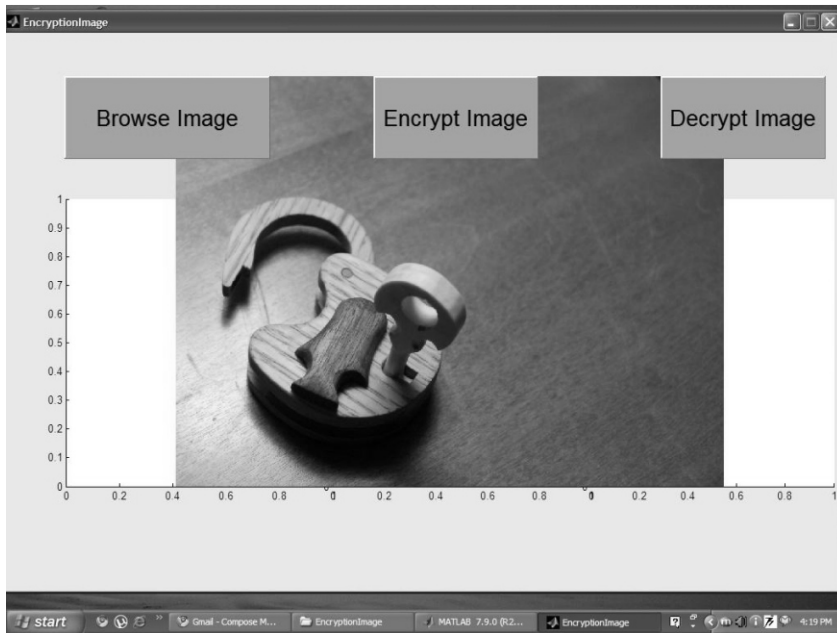


Figure XI: Front Page of the Interface



Figure XII : Interface after Browsing the Original Image

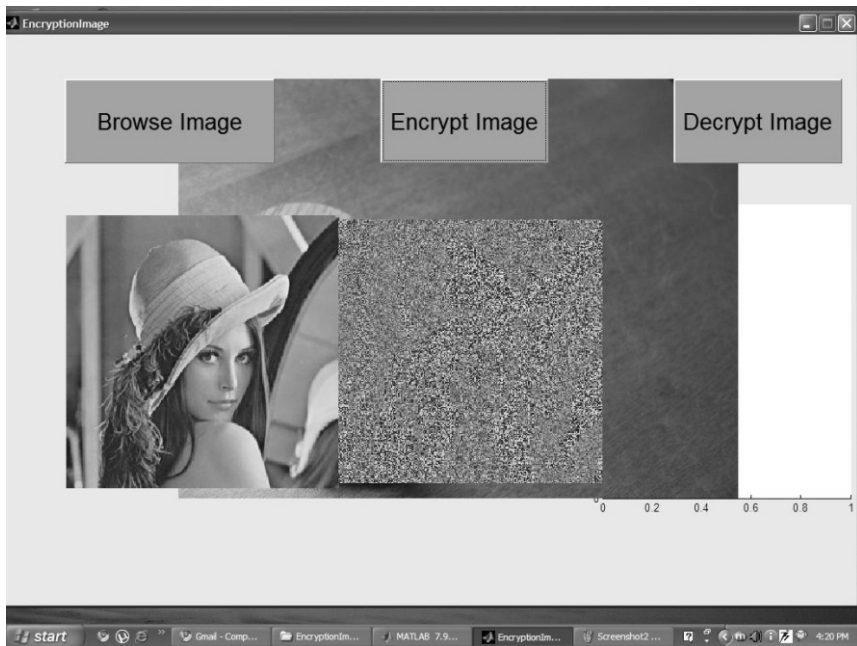


Figure XIII: Original and Encrypted Image after Encryption

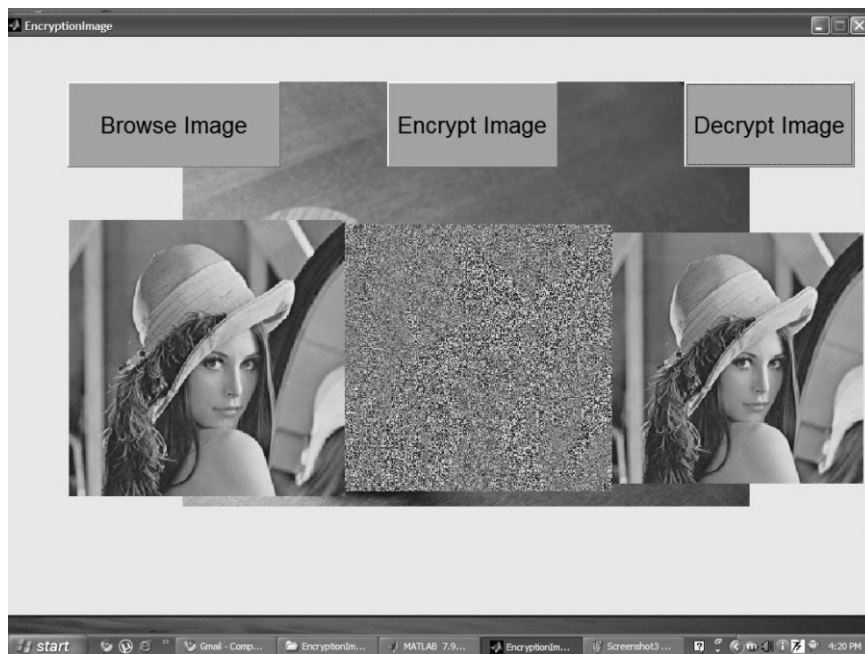


Figure XIV : Original Encrypted and Decrypted after Total Process Done


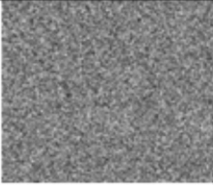





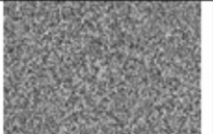


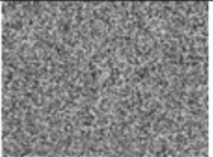


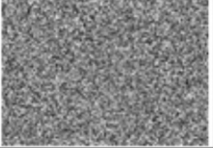

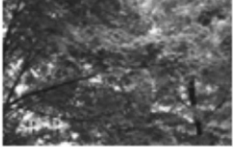
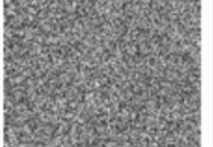


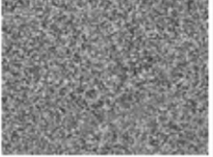

Sl. No	Original Image	Encrypted Image	Decrypted Image
01.			
02			
03			
04			
05			
06			
07			

Figure XV : Algorithm has been Tested on a Different Number of Bench-Mark or Non-Bench Mark Images are Furnished above