International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 5, May - 2013

An Approach Of Visual Cryptography Scheme For Color Image By Cumulative Encryption Using Image Partitioning, Text Key Encryption, Image Key Encryption & Digital Enveloping

Ramkrishna Das
Dept. of Computer Applications
Haldia Institute of Technology,
Haldia,WB,INDIA

Saurabh Dutta
Dept. of Computer Applications
Dr. B. C. Roy Engineering College
Durgapur-713206, WB, INDIA

Samarendra Kuila
Dept. of Computer Science
Vidyasagar University
Paschim Medinipur,WB, INDIA

Abstract

Visual Cryptography is a special type of encryption technique to obscure image-based secret information which can be decrypted by Human Visual System (HVS).A digital envelope is data container that is used to protect a message through encryption and data authentication. A digital envelope allows users to encrypt data with the speed of secret key encryption and the convenience and security of public key encryption.

At first this cryptographic system divide the secret image by partitioning it into n number of split images.In this current work we have proposed a variable lengthSymmetric Key based Visual Cryptographic Scheme for colorimages where a Secret key is used to encrypt the every images and the Encrypted images are again encrypted by the fingerprints images. Finally every encrypted images are enveloped in other images using invisible digital watermarking.

Keywords- *Digital Enveloping, Image key encryption, Image Partitioning, Text key encryption, Visual Cryptography.*

1. Introduction.

Cryptography is the practice and study of techniques for secure transmission of information between receiver and sender in the presence of other parties.
Visual cryptography is a cryptographic technique where visual information (Image, text, etc) gets encrypted in such a way that the decryption can be performed by the human visual system without aid of computers [1].
Like other multimedia components, image is sensed by human. Pixel is the smallest unit constructing a digital image. Each pixel of a 32 bit digital color image are divided into four parts, namely Alpha, Red, Green and Blue; each with 8 bits. Alpha part represents degree of transparency.
A 32 bit sample pixel is represented in the following figure [2] [3].

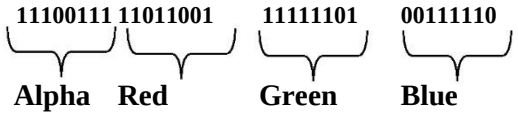


Fig 1: Structure of a 32 bit pixel

Human visual system acts as an OR function .The secret image is divided into four part.Then every part of the image are encrypted with different secret text key. Then cipher images are encrypted with finger prints images. Fingerprints are nothing but images. This type of visual cryptography technique is insecure as the reconstruction is done by simple OR operation.To add more security to this scheme we have proposed a technique called digital enveloping. This is nothing but an extended invisible digital watermarking technique.Using this technique, the split images are embedded into the envelope images by LSB replacement.The color change of the envelope images are not sensed by human eye.This technique is known as invisible digital watermarking as human eye can not identify the change in the envelope image and the enveloped (Produced after LSB replacement) image.In the decryption process k number of embedded envelope images are taken and LSB are retrieved from each of them followed by OR operation to generated the original image.
In this paper section-3 describes the encryption process; section-4 describes the decryption process. An experimental result is being described in section-5 and section-5 draws the conclusion.

2. Overall Process.

Step I: The source images is partitioned into n number of equal pieces.

Step II: Any combination of characters [Characters, Numbers and Special Symbol] of any length is taken as KEY, which is

XOR with the pixel array computed from a split image which is taken from step I. Does this process separately all split images from step I.

Step III: The encrypted image is encrypted again by fingerprint image. Fingerprints are images of separate finger of left hand of user's hand.Which isXOR ed with the pixel array computed from the encryptedimage. This makes the image more blur to some extent. Do this process until all encrypted images are again encrypted by several finger prints.

Step VII: The particular KEY taken in Step II is XOR with the particular image of n number of decrypted images produced in Step VI , to generate the Decrypted image.
Step VIII: n number of split decrypted images is produced in Step VII, is stackedtogether to reconstruct the image.
The process is described by Figure 1.

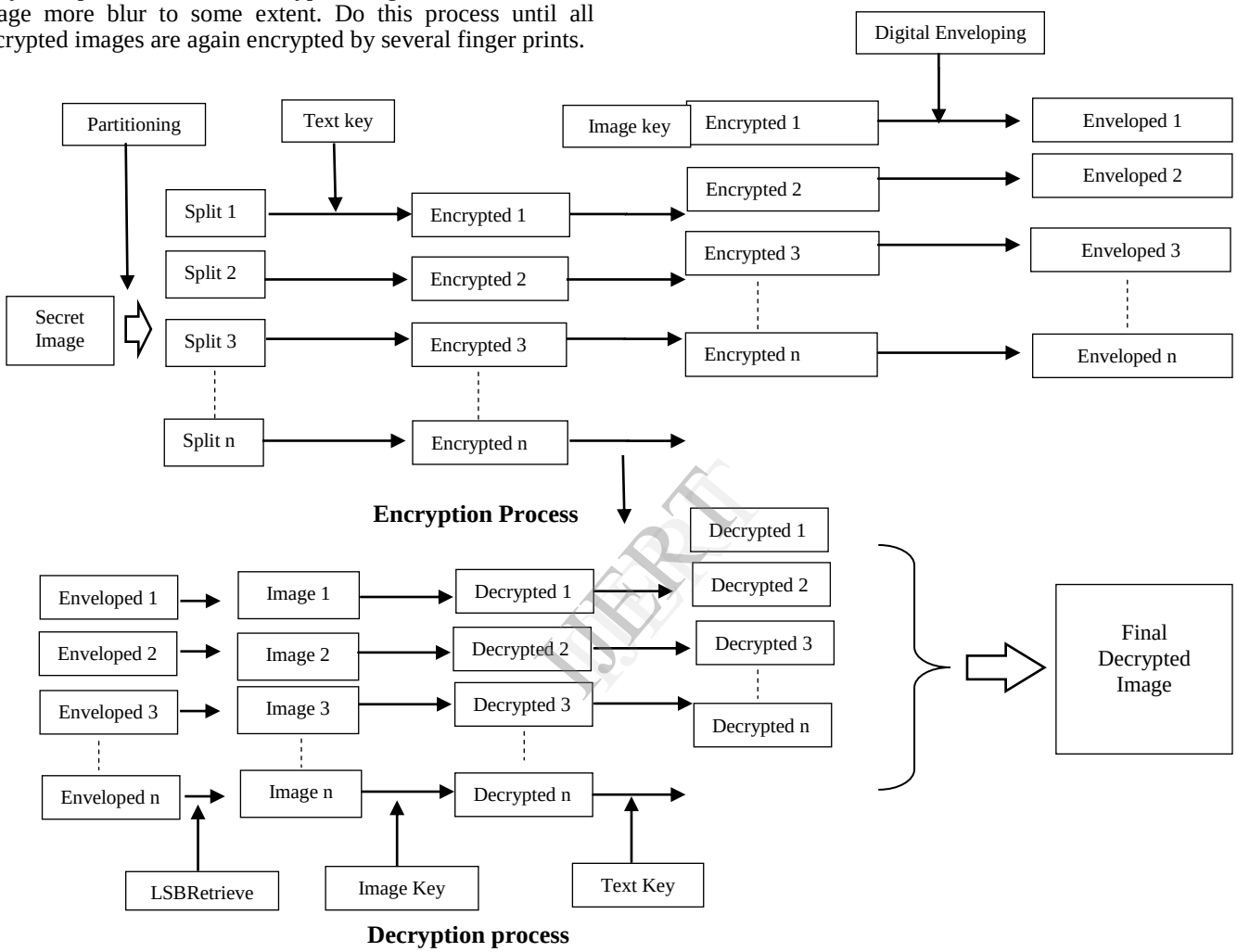


Fig 1: Overall Process

Step IV: Each of the images generated in Step III is embedded into n number of different envelope images using LSB replacement technique.

Step V: n number of enveloped images generated in step IV are taken and LSB retrieving, the split (each of same width and height) images are produced.

Step VI: Particular fingerprints image taken in step III is XOR ed with the particular an image of n number of split images produced in step V.

3. Encryption process.

Encryption process in following figure 2 represents 4 level of encryption. The levels of encryptions are image partition, text key encryption, image key encryption and digital enveloping. Figure-2 represents the entire procedure.

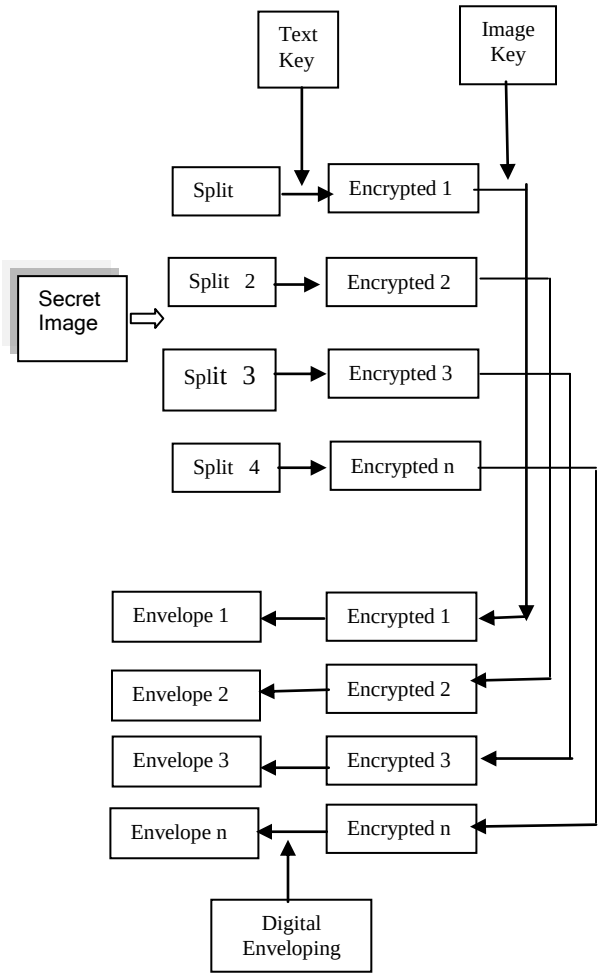


Figure 2: Encryption Process

Step-3.1:Image Partitioning

We take any secret image as inputwhich is partitioned by row and column. You can assume that this image is a table. Every table has number of rows and number of column. Every field of the table is being determined an image.

1.1 Taking an Image. Reading the image file.
1.2 Deciding the values for rows and cols variables.
1.3 Determines the image (chunk) width and height.
1.4 Taking Image array to hold image chunks and Initialize the image array with image chunks.
1.5 Draws the image chunk. Writing mini images into image files

Step-3.2: Text Key Encryption

One split image is taken as input from n no of images. A key of any length is taken as input from keyboard. An XOR operation is done on theimage using the key to generate the cipher image.Following algorithm is used for encryption.

Step I: Take an image as input. Calculate Height (h) and Width (w) of the image.
Step II: Create an array STORE of size w*h*32 to store the binary pixel values of the image using the loop
for i = 0 to (w*h-1)
{ Scan each pixel value of the image and convert it into 32

{ STORE[i*32+j] = PIX.charAt(j)
}
}
Step III: Enter a key of any length from keyboard. Calculate the length (len) of the key. Convert the key into binary string let CONVERTED_KEY.
[String is broken into character. Characters are converted into binary of length 7, then merged again to produce CONVERTED_KEY]
Step IV: Create an array KEY of size len*7 to store the binary values of the key by the following process.
for i = 0 to (len-1) {
KEY[i] = CONVERTED_KEY. charAt(i)
}
Step V: Calculate ITERATION = (w*h*32)/(len*7). KEY array is XOR ed with the STORE array by the following process.
for i = 0 to (ITERATION-1)
{ for j = 0 to (len*7-1)
{
STORE[i*len*7+j] = STORE[i*len*7+j]^KEY[j]
} //XOR operation
}
Encryption with Visual Cryptography
Decryption Process secret valuecumulatively. XOR operation is performed bitwise between the plain textand the secret image.

Step – 3.3: Image key encryption

The no of split images are encrypted by text key in step 2. Then the encrypted images are again encrypted by key images. Key images are fingerprints images which are unique individual images. An XOR operations are done
on the images using the key image to generate the cipher images.

Step I: Take an encrypted image(obtained from step 2) as input. Calculate Height (h) andWidth (w) of the image.

Step II: Create an array STORE of size w*h*32 to store the binary pixel values of the image using the loop

for i = 0 to (w*h-1)
{ Scan each pixel value of the image and convert it into 32 bit binary string let PIX
for j = 0 to 31
{ STORE[i*32+j] = PIX.charAt(j)
}
}

Step III : Take an fingerprints image as input and calculate Height (h) and Width (w) of the image.

Step IV: Create an array IMGKEY of size w1*h1*24 to store the binary pixel values of the image using the loop for l=0 to (w1*h1-1)
{ Scan each pixel value of the image and convert it into 24 Bit binary string let PIX1.(we take only RGB)
for k=0 to 23
{ IMGKEY[l*24+k]=PIXI charAt(k)
}
}

Step VI: Create a one dimensional array IMG_CONS[w*h] to store constructed pixels. Alpha remains same
Construct Red, Green and Blue part of each pixel by taking consecutive 8 bit substring starting from 0th bit.
Construct pixel from these part and store it into IMG_CONS [w*h].

Step VII: Generate image ENCRPT_IMG from IMG_CONS[w*h].

Step VIII: Repeat step I to Step VII until all encrypted images
In Step 2 are encrypted by several fingerprints images.

Step 3.4: Digital Enveloping

Using this step the finally n no. of encrypted images from step 3 of the original imageare enveloped within other image. Least Significant Bit(LSB) replacement digital watermarking is used for thisenveloping process. It is already discussed that a 32 bitdigital image pixel is divided into four parts namely alpha, red, green and blue; each with 8 bits. Experiment shows that if the last two bits of each of these parts are changed the changed color effect is not sensed by human eye[6]. This process is known as invisible digital watermarking [7]. For embedding 32 bits of a pixel of a divided share, 4 pixels of the envelope image is necessary. It means to envelope a share with resolution w X h; we need an envelope image with w X h X 4 pixels. Here we have taken each envelope of size 4w X h.
The following figure describes the replacement process. For replacing 8 bit alpha part, a pixel of the envelope is needed. In the same way red, green and blue part are enveloped in three other pixels of the envelope image.
The enveloping is done using the following algorithm
Step I: Take number of shares (n) as input.
for share = 0 to n-1 follow Step II to Step IV.
Step II: Take the name of the share, let SHARE_NO (NO is from 0 to n-1) and name of the envelope, let ENVELOPE_NO (NO is from 0 to n-1) as input. Let the width and height of each share are w and h. The width of the envelope must be 4 times than that of SHARE_NO.
Step III: Create an array ORG of size w*h*32 to store the binary pixel values of the SHARE_NO using the loop
for i = 0 to (w*h-1)
{ Scan each pixel value of the image and convert it into 32 bit
binary string let PIX
for j = 0 to 31
{ ORG [i*32+j] = PIX.charAt (j)
}
}
Create an array ENV of size 4*w*h*32 to store the binary pixel values of the ENVELOPE_NO using the previous loop but from i = 0 to 4*w*h*32 −1.
Step IV: Take a marker M= −1. Using the following process the SHARE_NO is embedded within ENVELOPE_NO.
for i = 0 to 4*w*h −1
{
Store value on 6^{th} and 7^{th} bit position.
Store value on 14^{th} and 15^{th} bit position.
Store value on 22^{th} and 23^{th} bit position.
Store value on 30^{th} and 31^{th} bit position.
}
Construct alpha, red, green and blue part of each pixel bytaking consecutive 8 bit substring starting from 0.
Construct pixel from these part and store it into a one dimensional array let IMG_CONS of size 4*w*h [].
}
Generate image from IMG_CONS [ ].

4. Decryption Process:

Here n numbers of enveloped images are taken as input. From each of these envelopes, we retrieve the last two bits of alpha, red, green and blue block of each pixel. Then encrypted pieces of image are reconstructed from these retrieved pixels. These encrypted pieces are decrypted by using image key first and the resultant encrypted pieces are decrypted by using text key. Then the final decrypted image is being reconstructed by using the decrypted pieces of the encrypted image. The overall process is represented in figure-3.

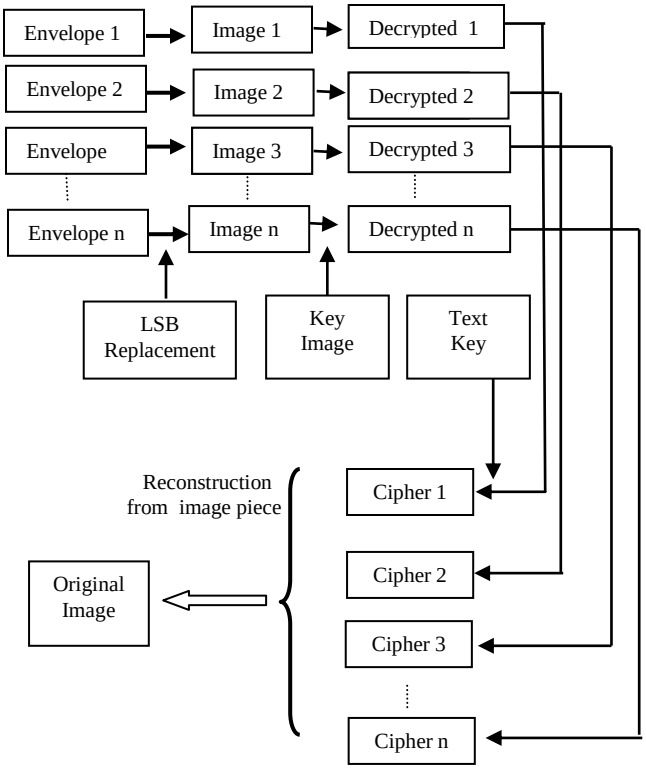


Fig 3: Decryption Process

Step 1 : Receiving enveloped images,are taken as input and calculate width(w) and height(h).
Create a two dimensional array STORE[k][w*h*32] to store the pixel values of n number of enveloped images. Create a one dimensional arrayFINAL [(w/4)*h*32] to store the final pixel values of each image which will be produced by taking the bit value retrieved from LSB position of each pixel from each enveloped images.

Step 2 :for envelope_no=0 to n-1
{
Take the name of the enveloped image to be taken and store the pixel values in STORE [envelope_no][w*h*32] using the following loop.
for i = 0 to (w*h-1)
{ Scan each pixel value of the Enveloped image and Convert it into 32 bit binary string let PIX.
for j = 0 to 31
{ STORE[Envelope_no][i*32+j] = PIX.charAt(j)
}
}
}

Step II: Take a marker M= −1. Using the following process the last two bits of alpha, red, green and blue block of each pixel of each n number of enveloped images and produce the pixels of the encrypted images piece.

Step III: A piece of encrypted image is taken as input and a image key is taken which was used for encryption for that particular piece. Then follow the process of step 3 of section 2 and generate the decrypted image by using image key decryption process. Do this similar activity for n numbers of images pieces produced by step II. Thus we get n numbers of decrypted image pieces generated by image key encryption. Dedicated image key which is used for encryption, that will be used for decryption for a particular image piece

Step IV: A piece of encrypted image is taken as input and a text key is taken which was used for encryption for that particular piece. Then follow the process of step 2 of section 2 and generate the decrypted image by using text key decryption process. Do this similar activity for n numbers of images pieces produced by step III. Thus we get n numbers of decrypted image pieces generated by text key encryption. Dedicated text key which is used for encryption, that will be used for decryption for a particular image piece

Step V :n number of split decrypted images is produced in Step IV, is stacked together to reconstruct the image by using simple or function. Thus the original image is reconstructed.

5. Experimental Results and Discussion

5.1 Encryption –

Partitioning using Visual Cryptography:
Source Image: cart.png
Source image is

Fig 4: Source Image
Number of split to be taken: 4
Image partitions produced after applying

Visual Cryptography:

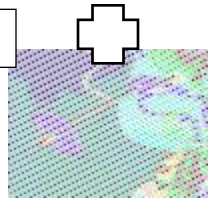

img0.png


img1.png


img2.png


img3.png

@mkll&d


img2.png


timage2.png

Text Key Encryption:


img0.png

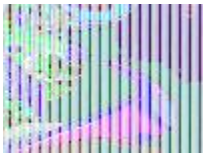
unknown@who


timage0.png

884rd

img3.png

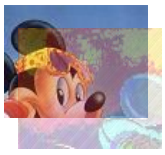
timage3.png



ssaammaress

img1.png


timage1.png

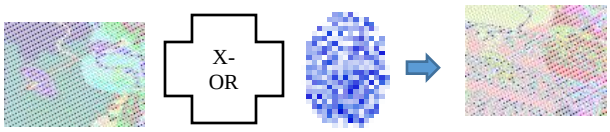


X-
OR

Image Key Encryption:

timage0.png f inger0.png fimage0.png

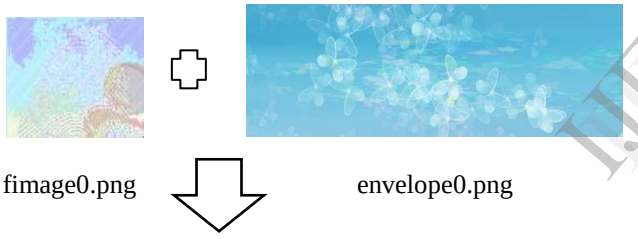
timage1.png finger1.png fimage1.png



timage2.png finger2.png fimage2.png

timage3.png finger3.png fimage3.png

Digital Enveloping:



fimage0.png envelope0.png

encrypted0.png

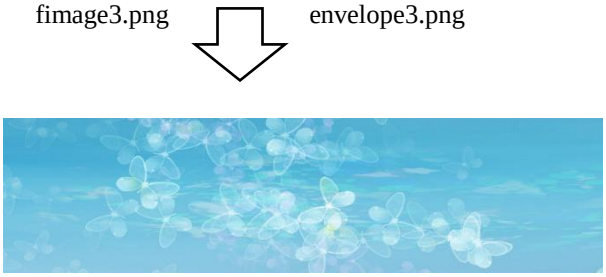
fimage1.png envelope1.png

encrypted1.png

fimage2.png envelope2.png

encrypted2.png

fimage3.png envelope3.png



encrypted3.png

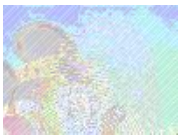
5.2 Decryption:

Denveloping:

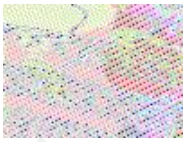
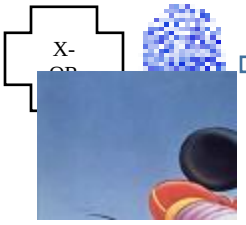
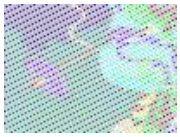
DeEnv0.png f inger0.png De4mImg0.png
DeEnv1.png finger1.png De4mImg1.png


encrypted0.png DeEnv0.png



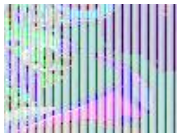
encrypted1.png DeEnv1.png

DeEnv2.png finger2.png De4mImg3.png

encrypted2.png DeEnv2.png

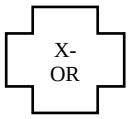




DeEnv3.png finger4.png De4mImg4.png

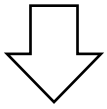
encrypted3.png DeEnv3.png

Text Key Decryption:

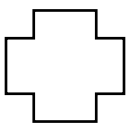
unknown@who

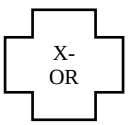
De4mImg0.png



De4mTxt.0.png

Image key Decryption:

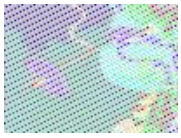



ssaammaress

De4mImg1.png

De4mTxt1.png



X-OR

@mkll&d

De4mImg2.png

De4mTxt2.png

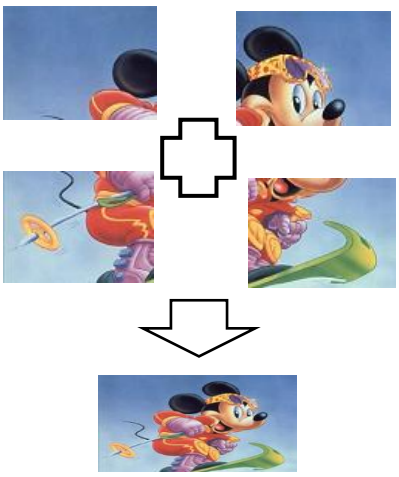
X-OR

884rd

De4mImg3.png

De4mTxt3.png

Image re-construction:



FinalImage.png

6. Conclusion

Decryption part of visual cryptography is based on OR operation, so if a person gets sufficient n number of shares; the image can be easily decrypted. In this current work we have increased the security by using several level of encryption procedure.

We have spited the image into several pieces so if anyone has to reconstruct the image then he must have to collect all the pieces which is very hard to collect. Thus the security is increased.

We have performed the text key encryption and image key encryption on each share .Text key and image key are dedicated for a particular piece and the keys are only being known by receiver and sender. Thus the security is increased.

The division of an image into n number of pieces is done by using the user choice, The decrypted image reconstruction is done by proper arrangement of the defined numbers of pieces. Thus provide a great security.

We have to send huge additional information of image key, text key, piece arrangement with the normal envelope image. Thus need additional memory space. In future we will focus on it and proposed some scheme which needs less amount of memory.

References:

[1] M. Naor and A. Shamir, "Visual cryptography," Advances in Cryptology-Eurocrypt'94, 1995, pp. 1–12.
[2] P. Ranjan, "Principles of Multimedia", Tata McGraw Hill, 2006.
[3] John F Koegel Buford, Multimedia Systems, Addison Wesley, 2000.
[4] KandarShyamalendu, MaitiArnab, "K-N Secret Sharing Visual Cryptography Scheme For Color Image Using Random Number" International Journal of Engineering Science and Technology, Vol 3, No. 3, 2011, pp. 1851-1857.
[5] Naskar P., Chaudhuri A, ChaudhuriAtal, Image Secret Sharing using a Novel Secret Sharing Technique with Steganography, IEEE CASCOM, Jadavpur University, 2010, pp 62-65.
[6] Hartung F., Kuttter M., "Multimedia Watermarking Techniques", IEEE, 1999.
[7] S. Craver, N. Memon, B. L. Yeo, and M. M. Yeung. Resolving Rightful Ownerships with Invisible Watermarking Techniques: Limitations, Attacks and Implications. IEEE Journal on Selected Areas in Communications, Vol16, No.4 May 1998, pp.573–586,.