International Journal of Innovative Technology & Adaptive Management (IJITAM)
ISSN: 2347-3622, Volume-1, Issue-7 , April, 2014

An approach of Visual Cryptography Scheme for Color Image by using Text Key Encryption & Image Key Encryption.

SK Usuf Ali
Dept. of Computer Science
Vidyasagar University
Paschim Medinipur, WB, INDIA
ali.usuf786@gmail.com

Ramkrishna Das
Dept. of Computer Applications
Haldia Institute of Technology,
Haldia, WB, INDIA
ramkrishnadas9@gmail.com

Saurab Dutta
Dept. of Computer Applications
Dr. B. C. Roy Engineering College
Durgapur-713206, WB, INDIA
saurabh.dutta@bcrec.org

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.
In this paper we have encrypted a image b y using both text and image key. Where the text key is used for encryption for every first and last bit of each eight bit block of a 32bit pixel and remaining bits of the pixels are encrypted by the image key. Here we have proposed a variable length symmetric Key based Visual Cryptographic Scheme for color image.

Keywords- *image key encryption, 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 figure 1 [2] [3].

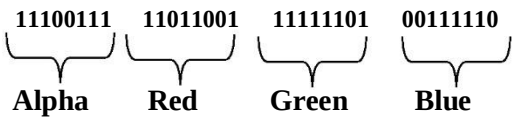


Fig 1: Structure of a 32 bit pixel

Take one pixel from the original image and convert it into four eight bit blocks (Alpha, Red, Green and Blue). First and last bit of eight bit block (Alpha, Red, Green and Red) block has been encrypted by the user inputted text key. All the other bits of the original pixel are been encrypted by the user inputted image key.
In this paper section-2 describes the overall process; section-3 describes the encryption process; section-4 describes the decryption process. An experimental result is being described in section-5 and section-6 draws the conclusion.

2. Overall Process.

Step I: The one pixel read from original image. All the pixels of the original image file are being converted and stored into OK [ ] where we take 32 bit representation for one pixel.
Step II: The 8 bit binary representation of each character for the input text key and 32 bit binary representation of each pixel are done. In this way all the characters of the text key and all the pixels of the image key are being converted and stored in TK [ ] and IK [ ] respectively.
Step III: Perform bitwise XOR between binary representation of the original image, text key and image key simultaneously.

CK [ ]: OK [I] ^ TK [P].
CK [ ]: OK [J] ^ IK [Q].

P = (0 to (number of char in text key * 8) - 1)
Q = (0 to (number of pixel in image key* 32) - 1)
CK [ ] = All pixels of cipher text image.
I = 0,7,8,15,16,23,24,31 for each pixel in original image.
J = 1-6, 9-14, 17-22, 25-30 for each pixel of original image.
Step IV: construct the encrypted image using array CK [ ] and send the encrypted image, text key, image key to the receiver
Step VII: The encrypted image would be taken as input. Convert the image into binary representation and perform bitwise XOR encrypted image, text key and image key. Now

construct the original image.
OK [ ] = CK [I] ^ TK [P].
OK [ ] = CK [J] ^ TK [Q].
The entire procedure is being described in figure 2.

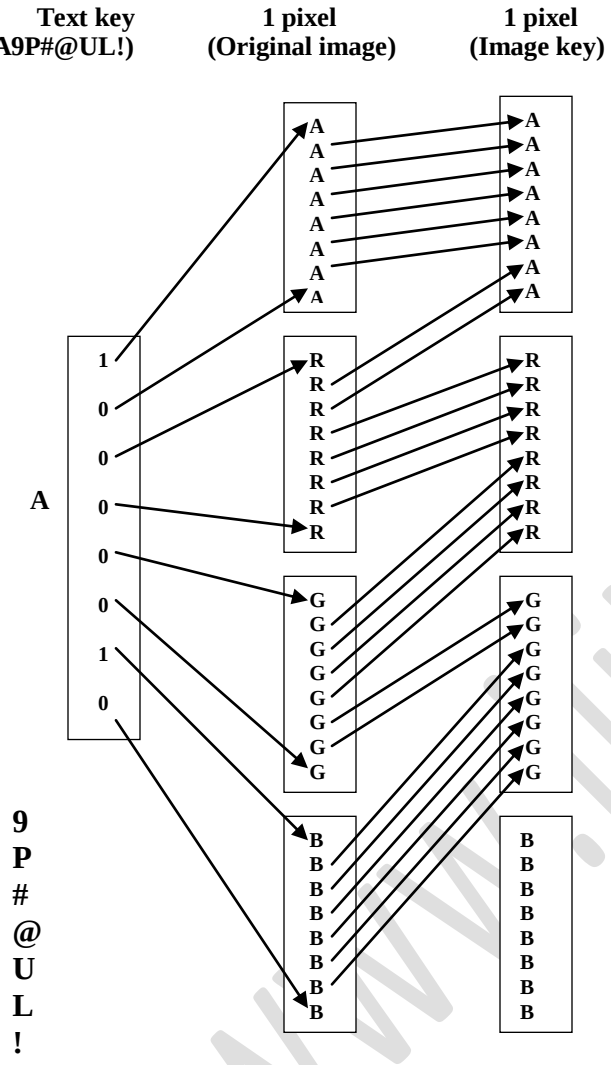


Fig 2: Overall Process

3. Encryption process.

Step I: The one pixel read from original image. All the pixels of the original image file are being converted and stored into OK [ ] where we take 32 bit representation for one pixel. The 8 bit binary representation of each character for the input text key and 32 bit binary representation of each pixel are done. In this way all the characters of the text key and all the pixels of the image key are being converted and stored in TK [ ] and IK [ ] respectively.

Step II: Perform bitwise XOR between binary representation of the original image , text key and image key simultaneously.

CK [ ]: OK [I] ^ TK [P].
CK [ ]: OK [J] ^ IK [Q].

P = (0 to (number of char in text key * 8) - 1)
Q = (0 to (number of pixel in image key* 32) - 1)
CK [ ] = All pixels of cipher text image.
I = 0,7,8,15,16,23,24,31 for each pixel in original image.
J = 1-6, 9-14, 17-22, 25-30 for each pixel of original image.

Construct the encrypted image using array CK [ ] and send the encrypted image, text key, image key to the receiver. Figure 3 represents the encryption procedure.

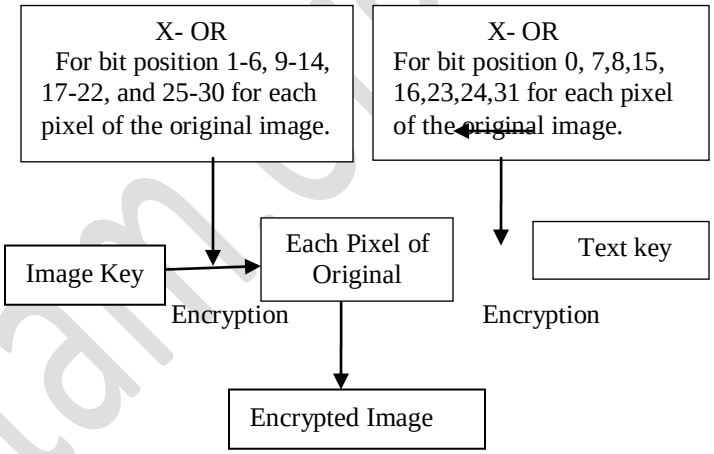


Figure 3: Encryption Process

The encryption algorithm is described in the section 3.1 and 3,2.

3.1: Text Key Encryption

One image is taken as input. A text key of any length is taken as input from keyboard. XOR operation is done on the binary representation of the original image using the text key to generate the cipher image. Following algorithm is used for encryption.

Step-I: Taking original image and text key as input.

Step-II: All the pixels of the original image file are being converted and stored into OK [ ] where we take 32 bit representation for one pixel. The 8 bit binary representation of each character for the input text key stored in TK [ ].

Step-III: Calculate the length original image (w1*h1) in bits and the text key size is defined to 8 characters (8*8=64 bits).

Step-IV: Perform XOR between OK [ ] and TK [ ] for bit position 0,7,8,15,16,23,24,31 for each pixel of the original image.

```
for (t=0; t<w1*h1; t++)
{    for (i=0; i<32; i++)
{  if (i==0||i==7||i==8||i==15||i==16||i==23
     ||i==24||i==31)
```

```
{   CK [(t*32) +i] =OK [(t*32) +i] ^TK [++p];
 If (p==63)
   p=-1;
}}
```

3.2 image key encryption

An XOR operation is done between the original image and image key for bit position 1-6, 9-14, 17-22and 25-30 for each pixel of the original image.

```
for (t=0; t<w1*h1; t++)
{   for ( i=0; i<32; i++)
{   If! (i==0||i==7||i==8||i==15||i==16||i==23
    ||i==24||i==31)
   {
  CK [(t*32) +i] =OK [(t*32) +i] ^IK [++p];
   if (p1== ((wik*hik*32)-1))
   p1=-1;
 }}}
```

Where w1*h1= size of original image and
Wik * hik = size of image key. Construct the encrypted image and send the receiver the encrypted image with the text and image key**.**

4. Decryption Process:

The encrypted image would be taken as input. Convert the image into binary representation and perform bitwise XOR encrypted image, text key and image key. Now construct the original image.

OK [ ] = CK [I] ^ TK [P].
OK [ ] = CK [J] ^ TK [Q].

P = (0 to (number of char in text key * 8) - 1)
Q = (0 to (number of pixel in image key* 32) - 1)
CK [ ] = All pixels of cipher text image.
I = 0,7,8,15,16,23,24,31 for each pixel in original image.
J = 1-6, 9-14, 17-22, 25-30 for each pixel of original image.
Construct the original image using array CK [ ]. Figure 4 represents the decryption procedure.

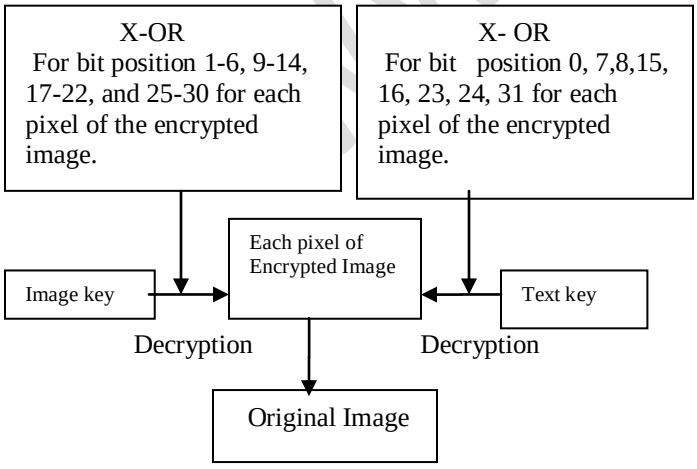


Fig 4: Decryption Process

The decryption algorithm is described in the section 4.1 and 4.2.

Encrypted image is taken as input. XOR operation is done on the binary representation of the encrypted image using the text key and image key to generate the original image. Following algorithm is used for encryption.

4.1: Text key Decryption.

Step-I: All the pixels of the encrypted image file are being converted and stored into CK [ ] where we take 32 bit representation for one pixel. The 8 bit binary representation of each character for the input text key stored in TK [ ].

Step-II: Calculate the length encrypted image (w1*h1) in bits and the text key size is defined to 8 characters (8*8= 64 bits).

Step-III: Perform XOR between CK [ ] and TK [ ] for bit position 0,7,8,15,16,23,24,31 for each pixel of the encrypted image.

```
 for (t=0; t<w1*h1; t++)
{   for (i=0; i<32; i++)
{
If (i==0||i==7||i==8||i==15||i==16||i==23||
    i==24||i==31)
  { OK [(t*32) +i] =CK [(t*32) +i] ^TK [++p];
    if(p==63)
      p=-1;
   }}
```

4.2 image key Decryption

An XOR operation is done between the encrypted image and image key for bit position 1-6, 9-14, 17-22and 25-30 for each pixel of the encrypted image.

```
for (t=0; t<w1*h1; t++)
{ for (i=0; i<32; i++)
{   if! (i==0||i==7||i==8||i==15||i==16||i==23
  ||i==24||i==31)
   {
      OK [(t*32) +i] =CK [(t*32) +i] ^IK [++p1];
       if (p1==(( wik*hik*32)-1))
        p1=-1;
  }
}
}
```

Where w1*h1= size of encrypted image and wik*hik= size of image key.

Generate the original image from OK [ ].

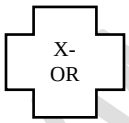
5. Experimental Results and Discussion

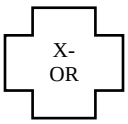
5.1 Encryption –

Source Image: lena.png
Source image is

Text key: usuf@ali
Image key: Sunset.png.
Image key is:

Image key
(sunset.png)





usuf@ali

Text key

Fig 5: Encryption Process

5.2 Decryption:

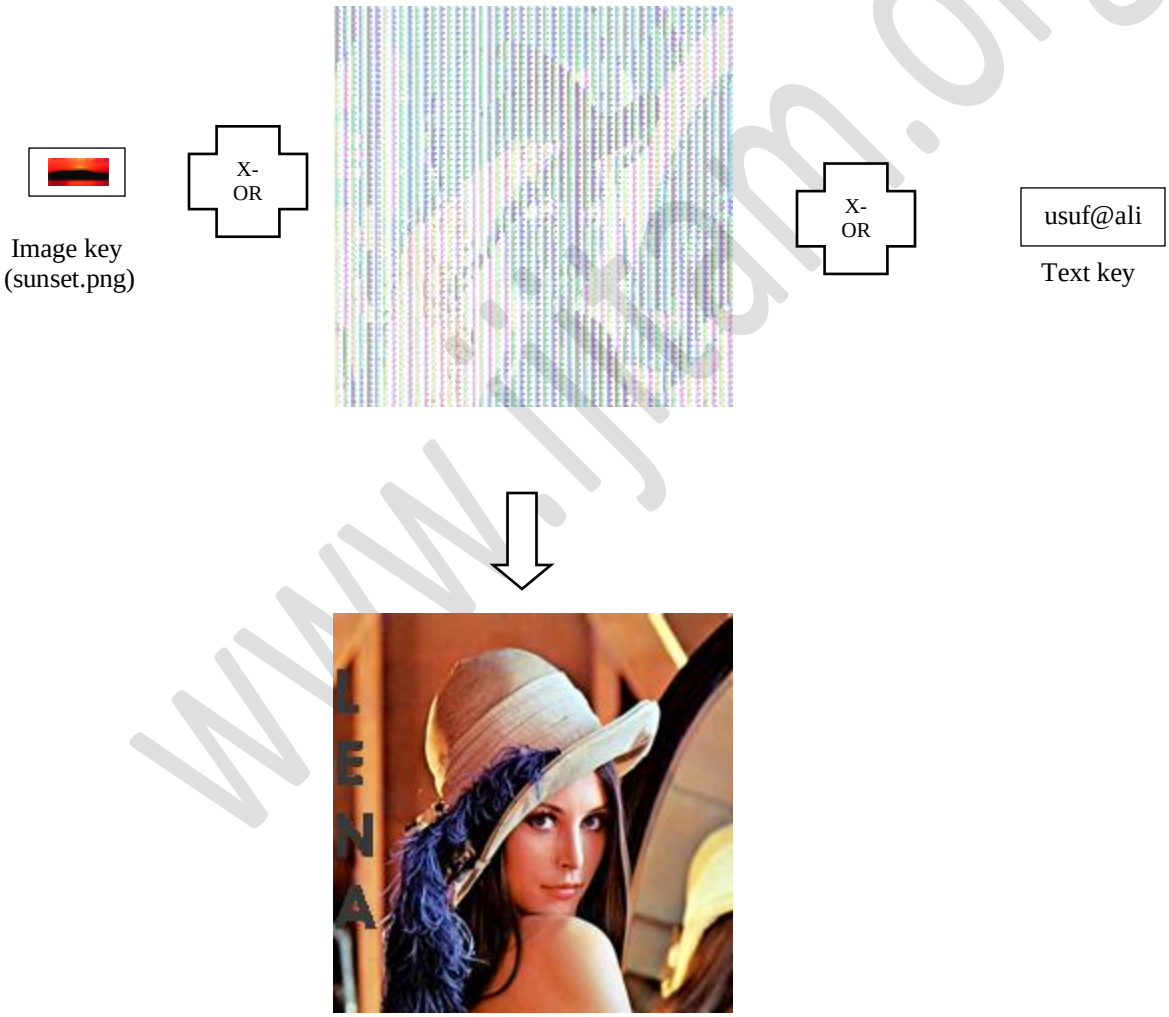


Fig 6: Decryption process

Watermarking Techniques: Limitations, Attacks and Implications. IEEE Journal on Selected Areas in Communications, Vol16, No.4 May 1998, pp.573–586,.

6. Conclusion

Decryption part of visual cryptography is based on XOR Operation, in this current work we have increased the security by using several level of encryption procedure.

We have both the text key and image key. Specific position is encrypted by the text key and image key simultaneously. The image key is size variable. Thus the security is increased.

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

At the time of XOR operation. How many times image key and text key will repeated, that is depend on the variable length size of the image key. Thus provide a great security.

We have to send huge additional information of image key, text key 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. Young. Resolving Rightful Ownerships with Invisible