International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-3 Issue-1, June 2013

An Approach of Visual Cryptography Scheme by Cumulative Image Encryption Technique Using Image-key Encryption, Bit-Sieved Operation and K-N Secret Sharing Scheme

Anupam Bhakta, Sandip Maity, Ramkrishna Das, Saurabh Dutta

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). It is imperceptible to reveal the secret information unless a certain number of shares (k) or more among n number of shares are superimposed. As the decryption process is done by human visual system, secret information can be retrieved by anyone if the person gets at least k number of shares. For this, simple visual cryptography is very in secure. In this current work we have proposed a method where we done the encryption in several level. First we use a variable length image key to encrypt the original image then bit sieve procedure is used on resultant image and lastly we perform K-N secret sharing scheme on the final encrypted image. Decryption is done in reverse level of encryption that means we do K-N secret sharing scheme, bit sieve method and image key decryption respectively. As multiple levels of encryptions are being used thus the security is being increased in great extant.

Keywords:- Bit Sieve Operation, Image Key Encryption, K-N Secret Sharing Scheme, Visual Cryptography.

I. INTRODUCTION

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] Image is a multimedia component sensed by human perception. A color digital image is composed of a finite number of elements called pixels. In a 32 bit digital image each pixel consists of 32 bits, which includes four parts, namely Alpha, Red, Green and Blue; each with 8 bits. Alpha part represents degree of transparency. If all bits of Alpha part are 0, then the image is fully transparent. A 32 bit sample pixel is represented in the figure1. [2] [3]

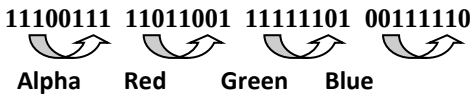


Figure 1: Structure of a 32 bit pixel

Manuscript published on 30 June 2013.
***Correspondence Author(s)**
Anupam Bhakta, Department of Computer Science, Vidyasagar university, Paschim Medinipur, W.B. , India.
Sandip Maity, Department of Computer Science, Vidyasagar university, Paschim Medinipur, W.B. , India.
Ramkrishna Das, Department of Computer Applications, Haldia Institute of Technology, Haldia , W.B. , India.
Saurabh Dutta, Department of Computer Applications, B.C. Roy Engineering College, Durgapur, W.B, India.

Human visual system acts as an OR function. If two transparent objects are stacked together, the final stack of objects will be transparent. But if any of them is non-transparent, then the final stack of objects will be non-transparent.

A image Key is used to make information secure so that attacker cannot retrieve the secret information without the image key. Original image is being encrypted using image key and then we do bit sieve operation on the encrypted image and finally we perform K-N secret sharing scheme generate cipher shares. Cipher shares are being taken and we perform K-N secret sharing scheme, bit sieve method and image key decryption respectively at the time of decryption.

In this paper Section 2 describes the Overall process, Section 3 describes the encryption process of the image, section 4 describes the decryption process, Section 5 describes the experimental result, and section 6 describes the conclusion.

II. OVERALL PROCESS

Step I: Any combination of characters (Characters, Numbers and Special Symbol) of any length is taken as KEY, which is XOR ed with the pixel array computed from the original image. This makes the image blur to some extent.

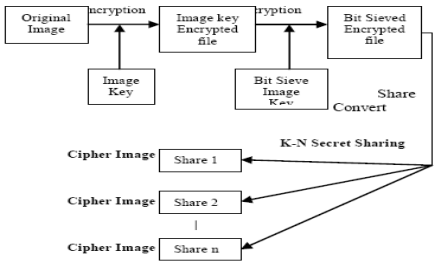
Step II: The encrypted image is bit sieved with the help of Image key and get bit sieve encrypted image.

Step III: The bit sieve encrypted image is divided into n number of shares using k-n secret sharing such that k number of shares is sufficient to reconstruct the encrypted image.

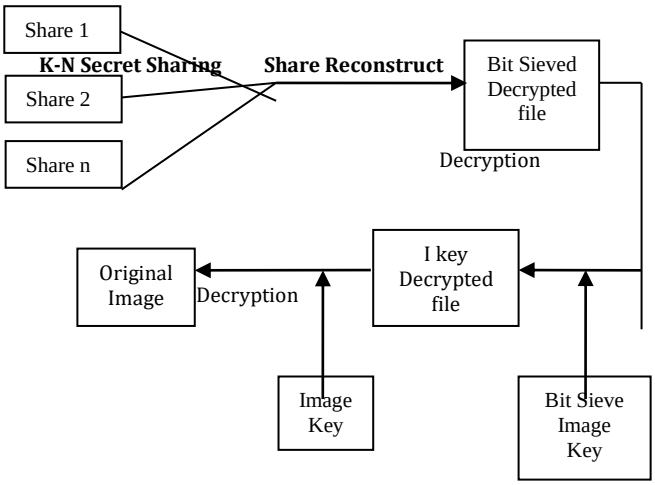
Step IV: k number of shares is stacked together to reconstruct the encrypted bit sieve image.

Step IV: The bit sieve encrypted image is reverse bit sieved with the help of Image key and get encrypted image.

Step V: The KEY taken in Step I is XOR ed with the image produced in Step IV, to generate the original image. This is described by the figure-2:



Encryption Technique.



Decryption Technique

Figure 2: Block Diagram of Overall Decryption Technique.

III. ENCRYPTION PROCESS

3.1 Image Key Based Encryption

An image is taken as an input. A key of any length is taken as input from the keyboard. An XOR operation is done on the image 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 bit binary string let PIX
for j = 0 to 31
{ 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
}

Step VI: Create a one dimensional array IMG_CONS[w*h] to store constructed pixels.
Construct Alpha, 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].1 [4][5]

Step VII: Generate image ENCRPT_IMG from IMG_CONS[w*h]. 3 [6]

3.2. Encryption by Bit sieved Procedure

Step I: Take an image as input and calculate its width (w) and height (h).

Step II: Create an array ORIGINAL of size w*h*32 to store the binary pixel values of the image using the following loop for i=0 to (w*h-1)
{Scan each pixel value of the image and convert it into32 bit binary string let PIX.
for j=0 to 31
{ if ith position of PIX contains '1'
Set that position value in ORIGINAL array to 1;}}

Step IV: Take an image as secret key and calculate its width (w1) and height (h1).

Step V: Create an array ORIGINAL_KEY of size w1*h1*32 to store the binary pixel values of the image using the loop for i=0 to (w1*h1-1)
{Scan each pixel value of the image and convert it into32 bit binary string let PIX1.
for j=0 to 31
{ if ith position of PIX1 contains '1'
Set that position value in ORIGINAL_KEY array to 1 }}

Step-VI Set key size= ORIGINAL_KEY.length();

Step-VII Dividing the image into two halves performing BIT SIEVE for each half individually with the key image by following the rule that if the key value is '0' then the original image value will be shifted to the 1st BIT SIEVE array otherwise the original image value will be shifted to the 2nd BIT SIEVE array for a particular bit position. Thus 4 BIT SIEVE array will be generated from two halves of original image.

Step-VIII Cross merging between two BIT SIEVE array will be performed to achieve the final BIT SIEVE array. Thus 2 final BIT SIEVE array will be generated.

Step-IX alternate sequence merging will be performed to generate the final BIT SIEVED Encrypted image array will be generated from where we can generated final BIT SIEVED Encrypted image .

Decryption Process can be performed by using a reverse method of the encryption algorithm.

3.3. k-n Secret Sharing Scheme based Encryption

The encrypted image obtained from Section 3, number of shares it will be divided (n) and number of shares to be taken to reconstruct the encrypted image (k) are taken as input. The division is done by the following algorithm.

Step: I: Take encrypted image ENCRPT_IMG produced in Section3 as input and calculate its width (w) and height (h).

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

Step II: Take the number of shares (n) and minimum number of shares (k) to be taken to reconstruct the image where k must be less than or equal to n. Calculate RECONS = (n-k)+1.
Step III: Create a three dimensional array IMG_SHARE[n][w*h][32] to store the pixels of n number of shares. k-n secret sharing visual cryptographic division is done by the following process.
for i = 0 to (w*h-1)
{
Scan each pixel value of ENCRPT_IMG and convert it into 32 bit binary string let PIX_ST.
for j = 0 to 31
{ if (PIX_ST.charAt(i) =1){
call Random_Place(n, RECONS)
}
for k = 0 to (RECONS−1)
{
Set IMG_SHARE [RAND[k]][i][j] = 1
}
}
}
Step IV: Create a one dimensional array IMG_CONS[n] to store constructed pixels of each n number of shares by the following process.
for k1 = 0 to (n-1)
{ for k2 = 0 to (w*h-1)
{ String value= “”
for k3 = 0 to 31
{
value = value+IMG_SHARE[k1][k2][k3]
}
Construct alpha, red, green and blue part of each pixel by taking consecutive 8 bit substring starting from 0.
Construct pixel from these part and store it into IMG_CONS[k1].1 [4]
}
Generate image from IMG_CONS[k1].3 [6]
}

IV. DECRYPTION PROCESS

The decryption process consists of into two steps. First step is done by human visual system where at least k number of shares out of n number of shares is superimposed, and the second step is decryption by the key taken in section 5 and the third step is decryption by the key taken in section 4. It is already discussed that human visual system acts as an OR function. For computer generated process; OR function can be used for the case of stacking k number of shares out of n.

4.1. Stacking k number of shares out of n:

k number of shares out of n is taken as input from user. As the shares are created from a single image in Section 4, each share is of equal height and width as the source image.
Bitwise OR operation is performed among pixels of the shares, and final pixel values are stored in an array. This is done by the following algorithm.
Step I: Input the number of shares to be taken (k); height (h) and width (w) of each share.
Step II: Create a two dimensional array SHARE [k][w*h] to store the pixel values of each share.
Create an one dimensional array FINAL[w*h] to store the final pixel values of the image which will be produced by performing bitwise OR operation. The OR operation is done by the following process. for i = 0 to k-1
{
Input the name of the i th image share to be taken.
for j = 0 to (w*h-1)
{
Scan each pixel value of the i th image share and store the value in SHARE[i][j].
}
}
Step III:
for i = 0 to (k-1)
{
for j = 0 to (w*h - 1)
{
FINAL[j] = FINAL[j] | SHARE[i][j];
} // [| is bitwise OR]
}

4.2 Reverse Bit sieved based Decryption:

Step I: The encrypted bit sieved image is taken as an input.
Step II: The image key used for bit sieve in encryption is taken as key.
Step III: Follow section 3.2 for reverse bit sieved operation and generates the decrypted image.

4.3 Image Key based Decryption:

Step I: The encrypted image from section 4.2 is taken as input.
Step II: The image key used in encryption is taken as key.
Step III: Follow section 3.1 for image key decryption and generate the final decrypted image.

V. EXPERIMENTAL RESULT

Image Key Encryption Process:
Key image: key1.png
Source image is

Figure 4 : Key Image

Source image: anupam.png
Source image is

Figure 5 : Source Image
Encrypted_ image is

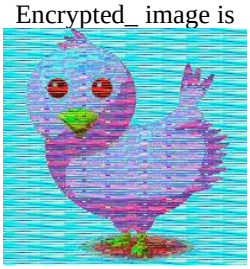


Figure 6 : Encrypted Image

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Bit Sieve Encryption Process:

Bit Sieve Key Image : apple.png
Bit Sieve Key Image is

Figure 7 : Bit Sieve Key Image

Bitsieved Encrypted image is

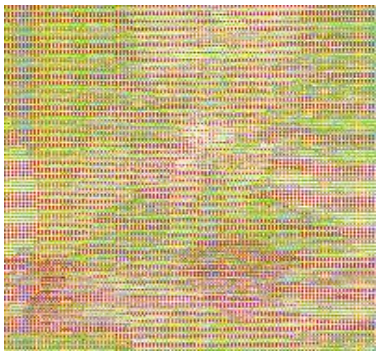
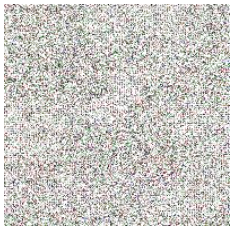


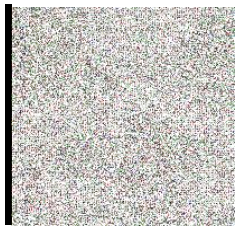
Figure 8 : Bit Sieved Encrypted Image

k-n Secret Sharing Visual Cryptography:

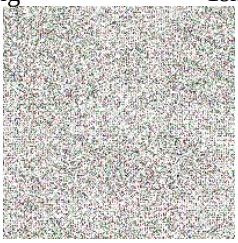
Number of Shares (n): 3
Numbers of shares to be taken (k): 3
Image shares produced after applying Secret Sharing:



0share.png



1share.png



2share.png

Figure 9 : Image shares produced after applying k-n Secret Shaering

VI. DECRYPTION PROCESS:

k-n Secret Sharing Visual Cryptography Decryption:
Number of shares: 2
Height and Width of each share: 200, 200
Shares inputted: 0share.png, 2share.png

The reconstructed image is

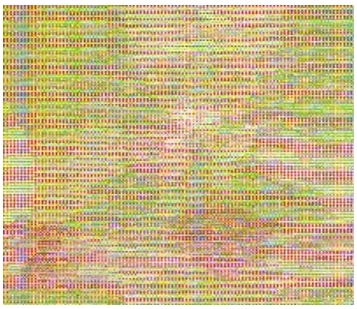


Figure 10 : Decrypted Bit Sieved Image

Bit Sieve Decryption Process:
Bit Sieve Key Image : apple.png
The Decrypted Image is

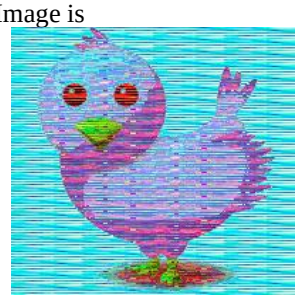


Figure 11 : Image Key Decrypted Image

Image Key Decryption Process:
Key image: key1.png
The Original Image is

Figure 12 : Final Reconstructed Image

VII. CONCLUSION

In this paper we have proposed a technique of well known image key based encryption, key based bit sieved encryption and k-n secret sharing based encryption. As we used multiple level of encryption thus the security is increased.
As image key and bit sieve based key are known in between sender and receiver so the security is increased.
We are taking specific K numbers of shares among N number of shares thus provide a more secured system.
Mathematical calculation compare with other existing techniques of secret sharing on color images [7][8][9] is very much less. So computation speed is increased.
But in this technique we have send huge additional information for image key based encryption, key based bit sieved encryption and k-n secret sharing scheme. So we need more memory. In future we try to eliminate this type of problem and make is more efficient.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

REFERENCES

1. M. Naor and A. Shamir, "Visual cryptography," Advances in Cryptology-Eurocrypt'94, pp. 1–12, 1995.
2. Ranjan Parekh, "Principles of Multimedia", Tata McGraw Hill, 2006
3. John F Koegel Buford, Multimedia Systems, Addison Wesley, 2000
4. Schildt, H. The Complete Reference Java 2, Fifth Ed. TMH, Pp 799-839
5. Krishmoorthy R, Prabhu S, Internet & Java Programming, New Age International, pp 234.
6. How to Split an Image into Chunks - Java ImageIO, http://kalanir.blogspot.com, Feb 2010
7. Naskar P.,Chaudhuri A, Chaudhuri Atal, Image Secret Sharing using a Novel Secret Sharing Technique with Steganography, IEEE CASCOM 2010, Jadavpur University pp 62-65
8. F. Liu1, C.K. Wu1, X.J. Lin, Colour visual cryptography schemes, IET Information Security, July 2008
9. Kang InKoo el. at., Color Extended Visual Cryptography using Error Diffusion, IEEE 2010