Hiding A Secret Image Into Two Hidden Images Using Visual Cryptography

Abu-Bakar Muhammad Abdullah, Nasif Ahmed

Abstract: Security has become a vital issue in the field of information technology. The study of Visual cryptography has attracted many researchers. It defines the division of a secret image into n shares from which the secret image cannot be revealed until all of the n shares are stacked together. A lot of innovations have been performed in this area such as hiding a secret image into two source images, recursive hiding of secret images, and extended visual cryptography for color images. In this paper, we introduce a technique for hiding three secret images using four source images. Two secret images can be revealed stacking two shares together. Finally, stacking all of the four shares together, the third secret image can be revealed.

Index Terms: Source image, secret image, shares, pixels.

1 INTRODUCTION

Cryptography ensures more security when the computations for the encryption and decryption process are more sophisticated. Unlike traditional cryptography, Visual cryptography doesn't need complex computations in decryption process. The basic idea of hiding information is based on the weakness of human perception. The encryption is performed in such a way that the human eyes cannot detect. The decryption is performed with the human eyes. For this reason to make the secret message more secured, the encryption process needs to be more complex. By this cryptographic technique we can encrypt visual information (pictures, text, etc.) in a way that human visual system can perform decryption of encrypted information and no aid of computers is needed. The most popular medium used is image files because of their high capacity and easy availability over the internet. In the visual cryptography, proposed by [1], each secret image is divided into two shares such that no information can be reconstructed from any single share. The decryption process is performed by superimposing the two shares and the secret image can be visualized by naked eye. Visual cryptography has many usage and applications such as digital signature, customer identification, watermarking, biometric authentication and remote electronic voting. As the visual information is available over the internet, it becomes necessary to hide them in order to protect from the cyber attackers. In the basic visual cryptography scheme each pixel of the secret image is extended to four pixels and it results the image which is 4 times the original secret image. In this paper, in order to increase the security, we are hiding the third secret image which is divided into two secret shares. These shares are hidden inside two secret images. That means, when two secret images are stacked together, the final image can be revealed. To do this each pixel is extended to six pixels. The proposed method can be applied in case of authorization of more than two persons. Besides it can be applied to increase the security of every third image as it needs all of the four shares.

- *Abu-Bakar Muhammad Abdullah, Lecturer, Department of CSE, Bangabandhu Sheikh Mujibur Rahman Science and Technology University, Gopalganj, Bangladesh, abdullahcse09@gmail.com*
- *Nasif Ahmed, Lecturer, Department of CSE, Bangabandhu Sheikh Mujibur Rahman Science and Technology University, Gopalganj, Bangladesh, ahmed.nasif08@gmail.com*

2 RELATED WORK

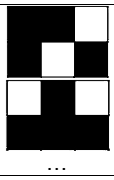
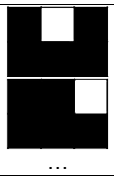
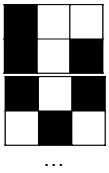
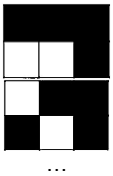
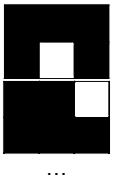
Several researches have been carried out in visual cryptography. Most of them are related to monochrome images. Nevertheless research on visual cryptography for color images is not lagging behind. The methodology by [1] paved the way for visual cryptography into deep exploration. New methodologies are coming forward with the torch of [1]. In the visual cryptography of [2], a secret information or image is split into n shares and that can be recovered by superimposing the shares. According to [3], additional information about small secrets is encoded in the shares of a larger secret without an expansion in the size of the latter. In [4], schemes of visual cryptography have been exploited to hide image and minor modification of extended visual cryptography has been proposed for color images. Multiple information hiding technique, proposed by [5] reveals one secret image stacking two shares normally and another secret image when one share is rotated 180º before stacking on other share. In the encryption process of [6], a stacking relationship graph of secret pixels and share blocks is generated to indicate the encryption functions, and a set of visual patterns is defined to produce two share images according to this graph. Five Modulus Method (FMM) is used in [7] to convert the pixels within both the cover and the stego images into multiples of five. A number from 1 to 4 is embedded secretly from the stego image inside the cover image using a 4x4 window. A method of image coding is proposed in [8] that hides the information along a selected pixel and on the next value of the selected pixel, that is, pixel+1. The 7th bit of the selected pixel and 7th bit of pixel+1 are used for information hiding and extraction. A novel reversible data hiding method based on neighboring pixel value differencing is proposed in [10] to increase embedding capacity with a good image quality.

3 PROPOSED METHOD

The concept of our methodology has some similarity with the visual cryptography proposed by [1]. Each pixel is being extended to six pixels. In order to create two shares of a secret image which are to be hidden inside two source images, the pixel of the secret image and the corresponding pixels of the source images are inspected. If the pixel of the secret image and the corresponding pixel of the source image are black, 67% is made black and remaining 33% is white. The white pixels are chosen randomly. If the pixel of the secret image is white and the corresponding pixel of the source image is black, 50% is made black. Simultaneously

we have to consider the same positioned pixel of another secret image along with the final secret image. The possible pixel representation for the share images, first 2 and final secret images are shown in the Table 1.

Table 1. Pixel representation for share and secret images

Pixel	Source image	Share image	Secret image #1, #2	Secret image #3
Black				
White				

Using the above patterns, we create the four shares taking all the three secret images into account. Though the patterns are to be chosen randomly, there are a number of cases where it becomes restricted to use a particular pattern. Some of the cases considered in the methodology are described below.

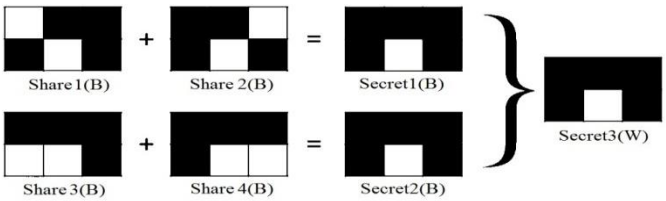


Fig. 1. When 4 source images have black pixel, secret images 1, 2 contain black but secret image 3 contains white pixel.

In Figure 1, the pixels of both the source images are black and their superimposition is to be black too. 2 pixels are chosen white for each source image such that only 1 pixel remains white after superimposition and its position is saved. In the next, if the pixels of the other source images and that of the second secret image are black whereas the third image has white pixel, 2 pixels are chosen white for each source image such that the saved positioned pixel is made white.

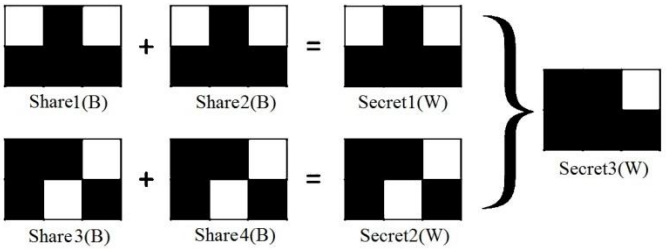


Fig. 2. When 4 source images have black pixel but secret images 1, 2 contain white and secret image 3 contains white pixel.

In Figure 2, the pixels of both the source images are black but their superimposition is to be white. 2 pixels are chosen white for each source image such that 1 pixel remains white after superimposition and its positions is saved. If the pixels of the other source images are black, that of the second secret image is white and the third image has white pixel, 2 pixels are chosen white for each source image such that the saved positioned pixel is made white.

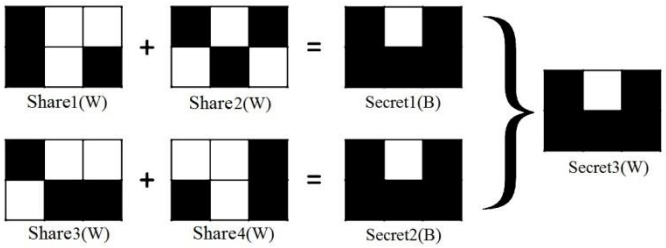


Fig. 3. When 4 source images have white pixel, secret images 1, 2 contain black but secret image 3 contains white pixel.

In Figure 3, the pixels of both the source images are white but their superimposition is to be black. 3 pixels are chosen white for each source image such that 1 pixel remains white after superimposition and its positions is saved. If the pixels of the other source images are white, that of the second secret image is black and the third image has white pixel, 3 pixels are chosen white for each source image such that the saved positioned pixel is made white.

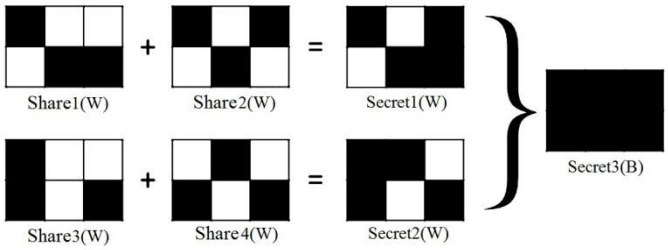


Fig. 4. When 4 source images have white pixel, secret images 1, 2 contain white but secret image 3 contains black pixel.

In Figure 4, the pixels of both the source images are white but their superimposition is to be white. 3 pixels are chosen white for each source image and 2 pixels which remain white after superimposition, 2 are saved. If the pixels of the other source images are white, that of the second secret image is white and the third image has black pixel, 3 pixels except the saved positioned pixels are chosen white for each source image such that the superimposition becomes black. The whole process, explained so far is shown below in Algorithm 1, 2 and 3. In Algorithm 1, 6 pixels for 2 shares are determined at a time. In Algorithm 2, at first, pixels of shares 1 and 2 are set then those of share 3 and 4. To perform this, we need to keep track of some pixels which must be white in all the 4 shares. Similarly, some pixels that are white must be remembered so that corresponding pixels can be made black in order to make 6 black pixels for the final secret image when they are stacked together. This tracking of pixel is said to be 'fixed' pixel in the algorithm.

First, in case of share 1 and 2, some pixels are kept fixed which can be helpful in the next operation for share 3 and 4. Finally, in Algorithm 3, the 4 shares are created completely.

Algorithm 1: SetPixels
Input: 4 source images, 2 secret images, (i, j) position
Output: required pixel pattern in a[] and b[]
/* a[] contains the pixel values for first share and b[] for second share */

```
if secretK(i, j) is black
    if sourceX(i, j) and sourceY(i, j) are black
        // when K = 1, X = 1 and Y = 2
        // when K = 2, X = 3 and Y = 4
        choose 2 pixels to be white for each share;
        if there is no 'fixed' pixel
            1 pixel position of one share is same as the other one;
            mark the common pixel position as 'fixed';
        else
            if secret3(i, j) is black
                4 pixels must not be same as 'fixed';
            else
                1 pixel in each share must be same as 'fixed';
    else if  sourceX(i, j) is black and sourceY(i, j) is white or sourceX(i, j) is white and sourceY(i, j) is black
        choose 2 and 3 pixels or 3 and 2 pixels to be white for shareX and shareY respectively;
        if there is no 'fixed' pixel
            1 pixel position of one share is same as the other one;
            mark the common pixel position as 'fixed';
        else
            if secret3(i, j) is black
                5 pixels must not be same as 'fixed';
            else
                1 pixel in each share must be same as 'fixed';
    else
        choose 3 pixels to be white for each share;
        if there is no 'fixed' pixel
            1 pixel position of one share is same as the other one;
            mark the common pixel position as 'fixed';
        else
            if secret3(i, j) is black
                6 pixels must not be same as 'fixed';
            else
                1 pixel in each share must be same as 'fixed';
else
    if sourceX(i, j) and sourceY(i, j) are black
        choose 2 pixels to be white for each share;
        if there is no 'fixed' pixel
            each pixel position of one share is same as the other one;
            mark the 2 common pixel positions as 'fixed';
        else
            if secret3(i, j) is black
                4 pixels must not be same as 'fixed';
            else
                1 pixel in each share must be same as 'fixed';
    else if  sourceX(i, j) is black and sourceY(i, j) is white or sourceX(i, j) is white and sourceY(i, j) is black
        choose 2 and 3 pixels or 3 and 2 pixels to be white for shareX and shareY respectively;
        if there is no 'fixed' pixel
            2 pixel positions of one share are same as the other one;
            mark the 2 common pixel positions as 'fixed';
        else
            if secret3(i, j) is black
                5 pixels must not be same as 'fixed';
            else
                1 pixel in each share must be same as 'fixed';
    else
        choose 3 pixels to be white for each share;
        if there is no 'fixed' pixel
            2 pixel positions of one share are same as the other one;
            mark the 2 common pixel positions as 'fixed';
        else
            if secret3(i, j) is black
                6 pixels must not be same as 'fixed';
            else
                1 pixel in each share must be same as 'fixed';
```

Algorithm 2: BuildShares
Input: 4 source images, 3 secret images, (i, j) position
Output: 4 secret shares

```
a[6], b[6];
SetPixels(source1, source2, secret1, secret3, i, j);
for all values of a[]
    set the pixels of share1 according to (i, j)th pixel position of source1;  /* 1 pixel of source1 is expanded to 6 pixels in share1 */
for all values of b[]
    set the pixels of share2 according to (i, j)th pixel position of source2;

SetPixels(source3, source4, secret2, secret3, i, j);
for all values of a[]
    set the pixels of share3 according to (i, j)th pixel position of source3;
for all values of b[]
    set the pixels of share4 according to (i, j)th pixel position of source4;
```

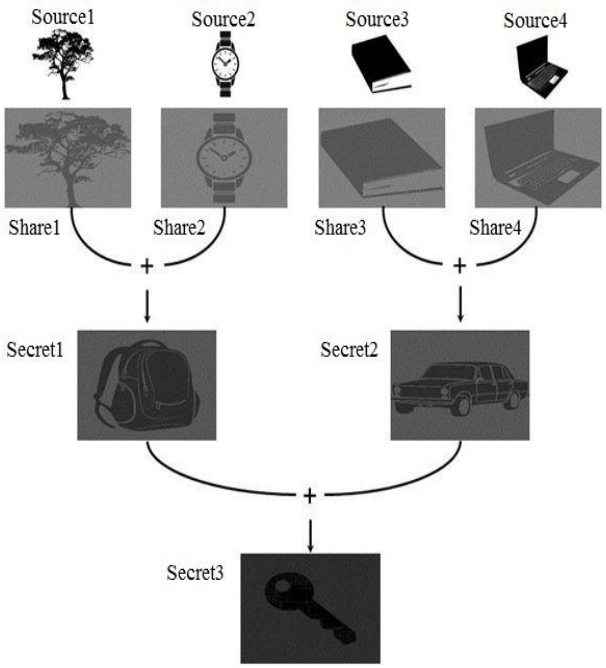
Algorithm 3: CreateShares

```
for all pixels positions (i, j) in mxn image
    BuildShares(4 source images, 3 secret images, i, j);
```

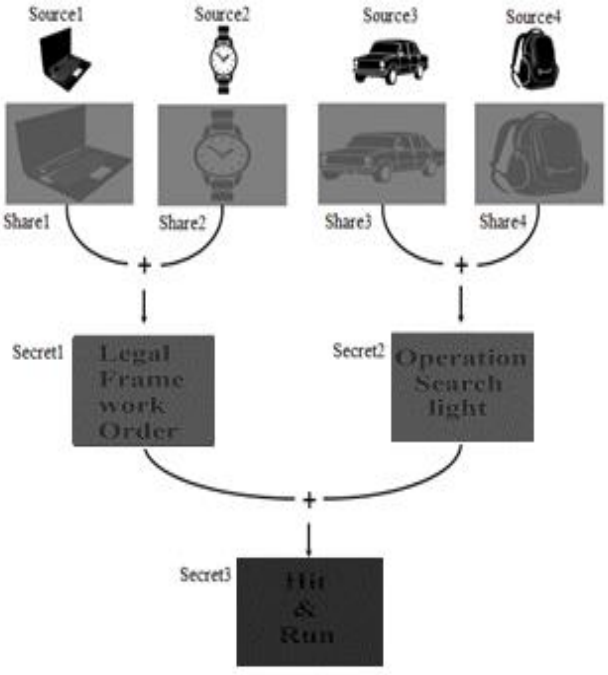
4 EXPERIMENTAL RESULTS

For each of source images and secret images, the binarization process is applied. The widths and heights of these images are made equal for simplicity. The following outputs are received when four source images and three secret images are fed into our proposed method. Four shares are the main outputs which contain the shares from all the three secret images. For this reason all the secret images can be revealed only when the output shares can be stacked in the right orderExperimental tests have been carried on some blank images as source and secret

images. Our proposed methodology also executes them correctly. Blank images do not cause any effect on the other images.



(a)



(b)

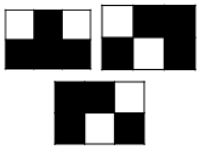
Fig. 5(a-b). Share1, Share2, Share3 and Share4 are the output images. When Share1 and Share2 are stacked together, Secret1 is revealed. When Share3 and Share4 are stacked together, Secret2 is revealed and when Secret1 and Secret2 are stacked, Secret3 is revealed.

In Figure 5, two experimental tests are illustrated. It is observed that the shares have nothing to be guessed about the secret image. When 2 shares are stacked together, the resulting image has no clue about the final secret image. This has also been evaluated in case of acute images that contain messages with some letters.

5 SECURITY ANALYSIS

In this section we discuss how easily the secret images can be revealed i.e. the approximate number of calculations needed for revealing the secret images. It is supposed that four shares are in possession of four different people. None can have more than single share. Let us consider that an impostor has two shares, Share1 and Share3 in his possession. Then how much hard work is needed for him to reveal the secret image? We have to calculate all the possibilities of the orientation of pixels. If pixels of Share1 or Share3 represent black, the numbers of possibilities for corresponding pixels of the secret image of being black and white are 2 and 1 respectively. When the pixels of Share1 or Share3 represent white, the numbers of possibilities for corresponding pixels of the secret image of being black and white are 3 and 3 respectively. The combinations are given in the Table 2. Let image size be m×n and there is average number of black and white pixels in Share1 or Share3. Therefore, for black representing pixels of Share1 or Share3, it needs $\frac{mn}{2}\times 3$ and that for white is $\frac{mn}{2}\times 6$. To reveal Secret1 or Secret2, the total number of calculations needed is (1.5mn)!(3mn)!. For each combination of Secret1 and Secret2, Secret3 can have a black or a white i.e. 2 possibilities. Hence, to reveal Secret3 it needs 2(1.5mn)!(3mn)!(1.5mn)!(3mn)!.

Table 2. Possible combinations of a secret image pixel

Pixel representing in Share1 or Share3	Pixel representing in Secret1 or Secret2	
	Black	White
		
		

6 CONCLUSION

Two source images are required for hiding one secret image according to visual cryptography proposed by [1]. Therefore, six source images are required for hiding three secret images. In this paper, we have introduced a method for hiding three secret images; four source images are required although total sizes of the images remains same. In [1], total size of the encrypted images is 6×(2m×2n) pixels for hiding three images if each image is of m×n pixels. In our proposed method, total size of the encrypted images is 4×(2m×3n) pixels. The cryptanalysis for first two

secret images is similar to [1] but for the third secret image it is more complex than [1].

REFERENCES

[1] M. Naor and A. Shamir, “Visual cryptography”, in Advances inCryptography-Eurocrypt, Vol. 950, pp. 1–12, Springer-Verlag (1995).

[2] A. Shamir, “How to Share a Secret”, in Communications of the ACM, Vol. 22, pp. 612-613 (1979).

[3] A. Parakh and S. Kak, “A Recursive Threshold Visual CryptographyScheme”, in Cryptology ePrint Archive (2008), Report 2008/535.

[4] Md. Tanbin Islam Siyam, Kazi Md. Rokibul Alam and Tanveer AlJami, “An Exploitation of Visual Cryptography to Ensure EnhancedSecurity in Several Applications”, in International Journal of Computer Applications, Vol. 65-No.6 (2013).

[5] B. Chhetri, S. Gurung, M.K. Ghose, Y.F. Chang, and Y.P. Chu, “Survey of Multiple Information Hiding Techniques using Visual Cryptography” in Compusoft, Vol. 4(1), pp. 1483 – 1488 (2014).

[6] J.B Feng, G.C. Wu, C.S. Tsai, Y.F. Chang, and Y.P. Chu, “Visual secret sharing for multiple secrets” in Pattern Recognition, Vol. 41, pp.3572 – 3581 (2008).

[7] F. A. Jassim, “Hiding Image in Image by Five Modulus Method for Image Steganography” in Journal of computing, 2013.

[8] K. Joshi , S. Gill, and R. Yadav, “A New Method of Image Steganography Using 7th Bit of a Pixel as Indicator by Introducing the Successive Temporary Pixel in the Gray Scale Image” in Journal of Computer Networks and Communications, 2018

[9] K. Joshi, R. Yadav, and S. Allwadhi, “PSNR and MSE based investigation of LSB,” in the International Conference on Computational Techniques in Information and Communication Technologies (ICCTICT), pp. 280–285, IEEE, New Delhi, India, March 2016.

[10] K. H. Jung, “Dual image based reversible data hiding method using neighboring pixel value differencing,” Imaging Science Journal, vol. 63, no. 7, pp. 398–407, 2015.