

PROPORTIONAL STUDY OF PIXEL SIEVE METHODS

Puneet Sharma

Department of Computer Science and Engineering,

Amity University, Uttar Pradesh

psharma9@lko.amity.edu

ARTICLE INFO

Received: 17 April, 2016

Accepted 15 May 2016

Corresponding Author:

Puneet Sharma

Department of Computer Science and Engineering, Amity University, Uttar Pradesh

Pradesh

Keywords: Visual Cryptography; Secret Sharing; Pixel Sieve Method; Improved Pixel Sieve Method.

ABSTRACT

The hasty innovation of network technology, multimedia information is communicated over the internet appropriately. Various confidential data such as military maps and commercial credentials are transmitted over the internet. To covenant with the security problems of secret images, various image secret sharing schemes have been established. Pixel Sieve Method is a method for encrypting secret images. This paper shows a comparative analysis of pixel sieve methods for visual cryptography.

©2016, IJICSE, All Right Reserved

1. INTRODUCTION

Pixel sieve methods are modern methods of visual cryptography. Visual Cryptography is a method to encrypt secret data without complex mathematical computations. In this technique secret data is kept in the form of an image and encryption process splits this image into shares. Secret data can be obtained by overlapping the splitted shares. Pixel sieve method is a technique which uses a Key Sieve to split the secret image into shares. In pictorial cryptography secret can be obtained by overlapping the shares only, while in Pixel Sieve Method secret cannot be obtained without key. Enhanced pixel sieve method is anticipated to advance its performance by familiarizing the "sieve and cross merge" and "key sieve shifting".

Here details of both methods and in later section a comparative analysis is presented with simulations.

2. Pixel Sieve Method

Pixel sieve method is a visual cryptographic method proposed by A.Incze. Pproposed pixel sieve method which uses a key to split the image. It is recycled to split a black and white image. This image is split in two shares both containing information about the main image but also cryptographic noise. The image is reconstructed from the shares not by overlapping the two shares, but applying a cryptographic process which also uses a key to reconstruct the original

image which will be visually interpreted afterwards. The reconstructed image will also contain some noise and slight deformations but in the limits of interpretability. The key used in this method is a binary image which contains holes like a sieve. The original image is placed over the key sieve. The pixels of the original image which are situated above the holes in the sieve go through and form one share. The enduring pixels form the other share of the image. The method is illustrated in the Fig. 1.

- The first line with black and light blue squares is the unique image.
- The sieve is publicized with dappled and white squares, where white squares are the holes.
- **Share 1** and **Share 2** are the two shares created.
- Arrows show how pixels move.

The method works as follows: we take the key image and the sieve pixel by pixel. If the assessment of the pixel in sieve is black then pixel from the main image goes to the **share 1** otherwise pixel goes to **share 2**.

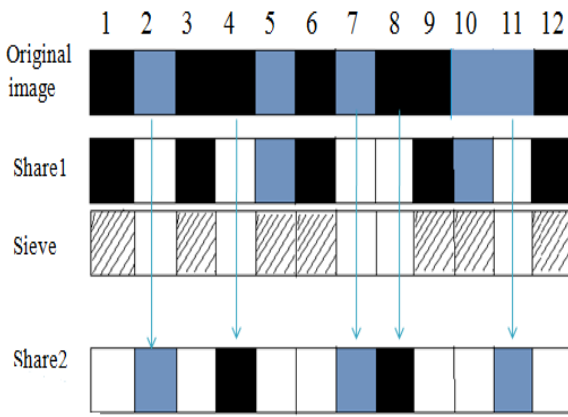


Figure 1: An example of pixel sieve method

There are two unlike approaches for creating segments from the image according to evolving in the shares.

1. For each advance in the original image there is also an advance in each share, whether the pixel is auxiliary to that share or not. (as in Fig. 1) In this method size of each segment is equal to the size of the image.
2. There is an advance in the share only when a pixel is added to that share from image. In this method shares have altered size and total size of both the segments is equal to the size of unique image. (as in Fig. 2)

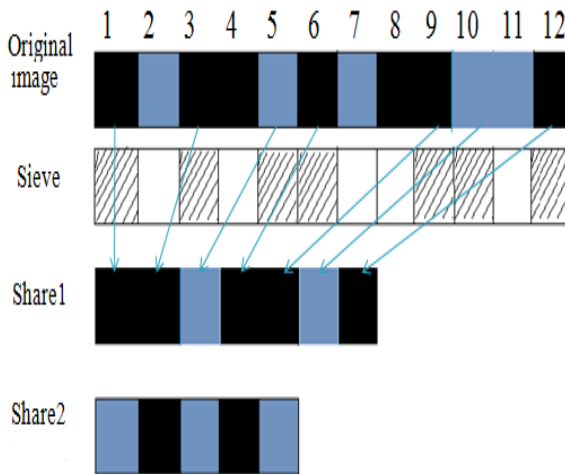


Figure 2: Compressed shares by second advancing method

Pixel sieve method is a dominant visual cryptographic procedure. It affords better security than older cryptographic methods, but it has some restrictions which can be solved by suggested methods. Original pixel sieve method has following precincts:

- a. In the first advancing method each share created is equal to the size of the image. Since the first advancing method produces two shares, the total size of the shares is twice of original image. Hence this method is very costly for large size images.

b. While in second advancing method each preceding pixel in the share also precedes in original image just after some pixels. If there are two pixels adjacent in a share they also neighbors in the original image with some pixels in between. The shares look such that some pixels are removed from original image and the image is compressed. The attacker can be able to perceive the secure data from single share in this method.

c. One major drawback of original pixel sieve method is that if we use a key slightly different from the key used in encryption to decrypt the image, we still gets some of the original data which is visually perceptible.

3. Improved Pixel Sieve Method

Improved Pixel Sieve Method is an advanced version of Pixel Sieve Method to overcome its limitations. The proposed method consists of two steps.

- a. Sieve and Cross Merge
- b. Key Sieve Shifting

Applying the cross merge and key shifting schemes the proposed method prevents the pixel expansion in encrypted image and enhances the security of the pixel sieve method. Moreover, we enhance pixel sieve method to reduce the chances for an attacker to guess the secret using keys which are nearly equal to the original key.

3.1 Sieve and Cross Merge

In the following section a modified sieving method is proposed, which removes the deficiencies of both the advancing methods. In this method first we split the secret image into two parts and apply the sieving process with first advancing method on each image part. In second step we merge the shares obtained from first step and produce two encrypted image parts. In last step both encrypted parts are joined together to create the encrypted image. The encrypted image produced in this method has the size equal to original image. We can iterate this method several times to enhance the security.

3.1.1 How It Works

The idea behind this method is based on an important property of the shares obtained by first advancing method of pixel sieve. In first advancing method when first share gets a pixel from original image, second share gets an empty pixel and when a pixel is added to second share, first share gets empty pixel. Hence both shares are the complement of each other. If we pixel-sieve two different images with same key sieve, then first share of first image is also a complement of second share of second image and first share of second image is a complement of second share of first image. Hence we can merge the first share of first

image with second share of other image and vice versa. The method is illustrated in Fig. 3.

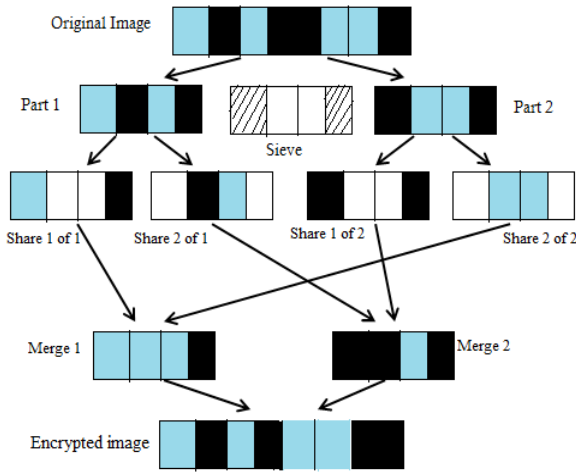


Figure 3: Sieve and cross merge process

Fig. 3.3 the **original image** is divided into two parts **Part 1** and **Part 2**. We pixel sieve both parts with same key sieve. In sieving process we get four shares **Share 1of1**, **Share 2of1** from **Part 1** and **Share 1of 2**, **Share 2of2** from **Part2**. Here **Share 1of1** has no data at **pixel 2 and 3**, while **Share 2of2** has data only on **pixel 2 and 3**. So we can merge both of them together. In same way **Share2of2** and **Share1of2** can also be merged. We get two encrypted image parts **Merge1** and **Merge2** by cross merging of the four shares. **Merge1** and **Merge2** are joined together in the last step to produce the final **encrypted image**. The way in which we joined the Merge1 and Merge2 is also important, because if we place them one after the other, we cannot iterate this process multiple times. If second iteration is performed on the encrypted image generated in first iteration, the original image is generated again. The merging process should be as follows:

First pixel of the encrypted image is taken from Merge1, second pixel from Merge2, third pixel again from Merge1 and so on.

The encrypted image produced in last is equal in size with original image. Hence this method does not increase the size of the image. Also, the encrypted image does not look like the compressed image. In this way, this method removes the problems of both advancing methods.

3.2 Key Sieve Shifting

In the original pixel sieve method each pixel of the key sieve encrypts only the corresponding pixel in the original image. Any pixel of key does not affect the encryption or decryption process of other pixels. Hence, if we use a key with some incorrect pixels to decrypt the image, only corresponding pixels will be decrypted incorrectly, while other pixels will be decrypted successfully. To remove this problem key

sieve shifting method is used. In this method we iterate the sieve and cross merge method several times with different shifted keys on the original image. We shift the key in each round of encryption process. In the decryption process the keys are used in reverse order of encryption process.

3.2.1 How It Works

Shifting of the encryption key is an important part of various cryptographic algorithms. In this method the key sieve used for pixel sieving is shifted in each round. We propose a key shifting method with two steps.

1. In first step we **circularly left** shift each row of the key sieve independently. Pixels of each row are shifted **n** times (**n** is equal to the number of **black pixels** in that row). Each row is shifted with different amount according to the number of black pixels in that row (Fig. 4).
2. In second step each column of the key is **circularly up** shifted independently according to the number of black pixels in that column (Fig. 5).

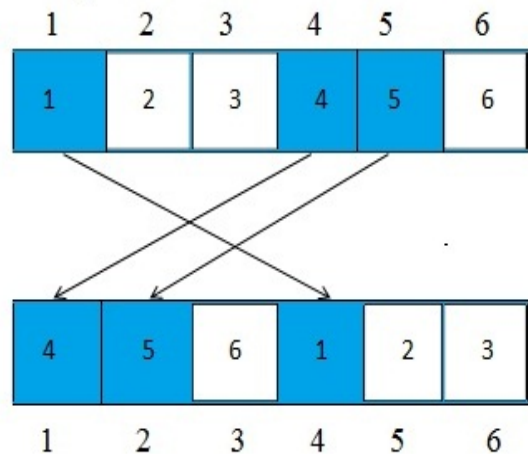


Figure 4: Shifting of a single row

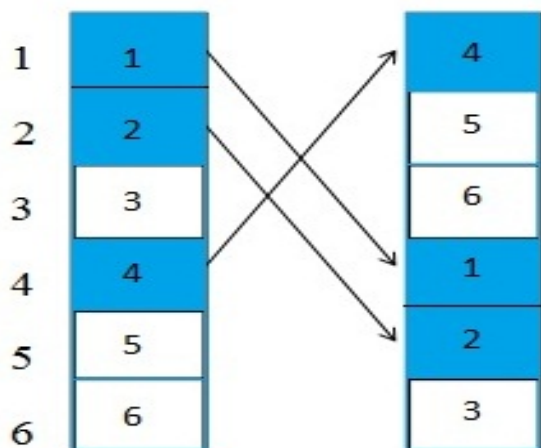


Figure 5: Shifting of a single column

After applying both the steps we get the shifted key. This shifted key is used for pixel sieving the image. In the next round this key is again shifted with the same

procedure and used in pixel sieving process. In this process the pixels of the key sieve move to different locations in each iteration of sieving process. Hence every pixel of the key sieve is involved in the encryption of the different pixel in different iterations. If any pixel of decryption key is incorrect then the whole row and column related to that pixel decrypted incorrectly. If any pixel of decryption key is different than the number of black pixels in that row is also different and the row is shifted with different amount rather than actual required shifting. If any shifting is incorrect it affects all the later key shifting, because due to incorrect shifting of any row, each column gets incorrect pixels and hence each column is shifted incorrectly in next iteration.

Key sieve shifting method enhances the security of the pixel sieve method. This method provides security against nearly equal keys used for decryption. Another advantage of this method is that it also increases the randomness in the decrypted image. The pixels in the encrypted image are scattered more randomly than the existing pixel sieve method.

4. Algorithm of Improved Pixel Sieve Method

Algorithms used for implementation of Improved Pixel Sieve Method are as follows:

4.1: Algorithm For Sieving Method

```

1. Int Image[m][n], Key[p][q], Share1[p][q],
   Share2[p][q], Share3[p][q], Share4[p][q];
2. Int Temp;
3. for(int i=0; i < p; i++)
4. {
5.     for(int j=0; j < q; j++)
6.     {
7.         Temp = Key[i][j];
8.         if(Temp == 0)
9.         {
10.            Share1[i][j] = Image[i][j];
11.        }
12.        else
13.        {
14.            Share2[i][j] = Image[i][j];
15.        }
16.    }
17. }
18. for(int i=0; i < p; i++)
19. {
20.     for(int j=0; j < q; j++)
21.     {
22.         Temp = Key[i][j];
23.         if(Temp == 0)
24.         {
25.            Share3[i][j] = Image[i+p][j];
26.        }
27.        else

```

```

28.        {
29.            Share4[i][j] = Image[i+p][j];
30.        }
31.    }
32. }

```

4.2: Algorithm For Merging Method

```

1. Int Image[m][n], Key[p][q], Share1[p][q],
   Share2[p][q], Share3[p][q], Share4[p][q], Merg1[p][q],
   Merg2[p][q];
2. Int Temp;
3. for(int i=0; i < p; i++)
4. {
5.     for(int j=0; j < q; j++)
6.     {
7.         Temp=Key[i][j];
8.         if(Temp==0)
9.         {
10.            Merge1[i][j] =
Share1[i][j];
11.        }
12.        else
13.        {
14.            Merge1[i][j] =
Share4[i][j];
15.        }
16.    }
17.     for(int j=0; j < q; j++)
18.     {
19.         Temp=Key[i][j];
20.         if(Temp==0)
21.         {
22.            Merge2[i][j] =
Share3[i][j];
23.        }
24.        else
25.        {
26.            Merge2[i][j] =
Share2[i][j];
27.        }
28.    }
29. }
30. for(int i=0,k=0;i<180;i++)
31. {
32.     for(int j=0;j<200;j++)
33.     {
34.         Image[k][j] = Merge2[i][j];
35.         Image[k+1][j] = Merge1[i][j];
36.     }
37.     k=k+2;
38. }

```

4.3: Algorithm For Key Shifting

```

1. Int BPixel, Temp, Key[p][q], NewKey[p][q];
2. for(int i=0; i < p; i++)
3. {

```

```

4.   for(int j=0;j< q;j++)
5.   {
6.   if(Key[i][j] == 1)
7.   {
8.       BPixel++;
9.   }
10.  for(int m=0;m<q;m++)
11.  {
12.      Temp=(m+BPixel)%200;
13.      NewKey[i][Temp] = Key[i][m]
14.  }
15.  }
16.  for(int j=0; j< q; i++)
17.  {
18.      for(int i=0;i< p; i++)
19.      {
20.          if(Key[i][j] == 1)
21.          {
22.              BPixel++;

```

```

23.      }
24.      for(int m=0; m<200;m++)
25.      {
26.          Temp=(m+BPixel)%200;
27.          NewKey[Temp][j] = Key[m][j]
28.      }
29.  }

```







5. Simulation of Both Methods

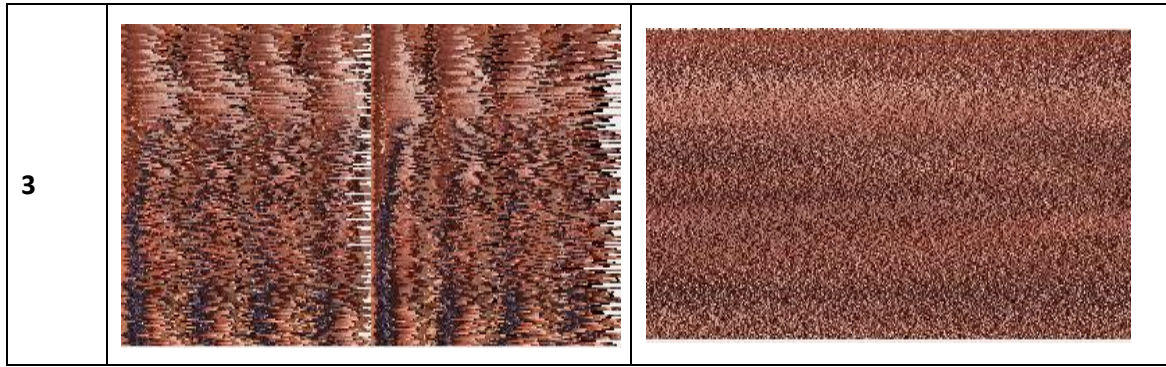
In this section we present extensive simulation results obtained by applying both older Pixel Sieve Method and Improved Pixel Sieve Method on same sample image(Fig. 4.2). Both methods are tested with different number of iterations and results are captured.

Table 4.1 shows the simulation results of both methods with increasing number of iterations.

5.1 Simulation Table for Encryption

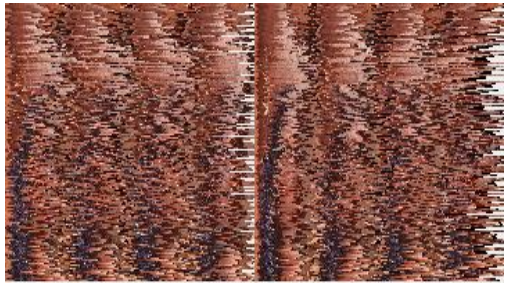

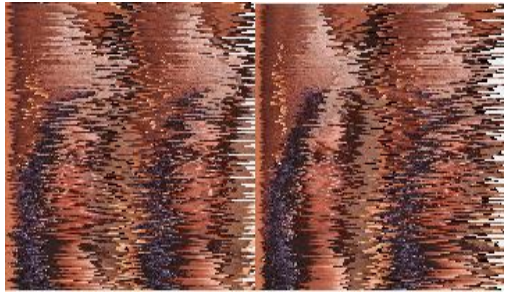

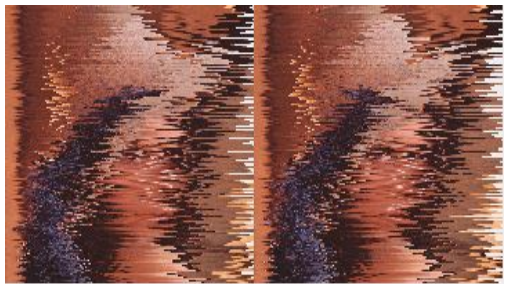



Table 1: Simulation Results for Encryption

Iterations	Pixel Sieve Method	Improved Pixel Sieve Method
0		
1		
2		



5.2 Simulation Table for Decryption

Table 2: Simulation Results Decryption

Iterations	Pixel Sieve Method	Improved Pixel Sieve Method
0		
1		
2		
3		

6. Conclusion

Simulation results show that encrypted images generated by Improved Pixel Sieve method are quite noisy and of high security. The sieve and cross merge scheme prevents the pixel expansion in encrypted image and enhance the security of the existing pixel sieve method and key sieve shifting method is used to reduce the chances for an attacker to guess the secret using keys which are nearly equal to the original key. The new method can be broadly used in a number of visual secret sharing applications which requires high quality secret images and high security such as electronic cash, secret maps etc.

References

1. Moni Naor and Adi Shamir. Visual Cryptography, EUROCRYPT 1994, ppl- 12
2. Shamir, Adi. "How to share a secret". Communications of the ACM 22 (II): 1979,6 12- 613
3. A.Incze, "Pixel Sieve method for secret sharing & visual cryptography". 9th RoEduNet IEEE International Conference 2010
4. P.S.Revenkar, Anisa Anjum, W .Z.Gandhare. " Survey of Visual Cryptography Schemes". International Journal of Security and Its Applications ,Vol. 4, No. 2, April, 2010
5. Pravin Kumar, Kishore Kumar, Vaibhav choudhary, D.S. Singh, "Modified pixel sieve method for visual cryptography" Indian Journal of Computer Science and Engineering Vol. 1 No. 4, December 2010
6. M.Naor, B.Pinkas, Visual authentication and identification. Advances in Cryptology-CRYPTO'97 LectureNotes in Computer Science, Vol. 1294, pp. 322-336
7. Shang-Kuan Chen, Sian-Jheng Lin. "Non-expansible Flip-flop Visual Cryptography with perfect security". 2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing
8. Frank Y. Shih " Digital Watermarking and Steganography: Fundamentals and Techniques" , CRC Press, 2007, ISBN: 978 1420047578
9. Chin-Chen Chang, Min-Shian Hwang, Tung-Shou Chen, "A new encryption algorithm for image cryptosystems", The Journal of Systems and Software 58 (2001), 83-9 1
10. Vaibhav choudhary, Kishore Kumar, Pravin Kumar, D.S. Singh, "An Improved pixel sieve method for visual cryptography" International Journal of Computer Applications Vol. 12 No. 9, January 2011
11. Ching-Sheng Hsu and Shu-Fen Tu, "Digital watermarking scheme with visual cryptography," in Proceedings of the International Multi Conference of Engineers and Computer Scientists IMECS 2008 Vol I, March 19–21, 2008
12. Amir Houmansadr and Shahrokh Ghaemmaghani, "A novel video watermarking method using visual cryptography," IEEE International Conference on Engineering of Intelligent systems, Islamabad, Pakistan, 2006