Image Encryption based on Inter Pixel Displacement of RGB Values inside Custom Slices

Amnesh Goel
Amity School Of Engineering
& Technology, Amity University,
Noida (U.P.), India

Reji Mathews
Amity School Of Engineering
& Technology, Amity University,
Noida (U.P.), India

Nidhi Chandra
Amity School Of Engineering
& Technology, Amity University,
Noida (U.P.), India

ABSTRACT

The encryption methods for enhancing the security of both text and multimedia contents has gained high significance in the current era of breach of security and misuse of the confidential information intercepted and misused by the un- authorized parties. Here we proposed an enhancement to an existing algorithm proposed early in which the RGB attributes of a pixel were randomly scattered across the image. The scattering algorithm works on each of the Red, Green and Blue pixel values and breaks each of the pixel with respect to its constituent pixel attributes and scatters them across the spatial space of the image thus making it difficult to reform the original image unless each of the R G B attribute of the pixels are located and identified to which spatial coordinate they belong to. In our enhancement, we proposed a technique to add further confusion property in the ciphering of image by slicing the image into n number of sub images whose dimensions are kept confidential and applying the above algorithm to each of these sub images. The sub images are then shuffled so as to further add to resistance towards the deciphering attacks.

General Term

Digital Image Processing.

Keywords

Image Encryption, RBG, Shifting, slicing, shuffling, permutation.

1. INTRODUCTION

The image encryption techniques have gained a very high popularity in transmission of sensitive images like those generated for the defense organizations which may contain images of strategic planning details and geographical location images. As we see the huge growth in the use of internet globally, the transmission of multimedia files such as videos, sound clips and images have increased manifolds with wider bandwidth being made available with cutting edge technology at cost effective rate. The information being sent over internet which is a very vulnerable public medium to which any individual has direct access, certain precautions have to be adopted in handling the transfer of the sensitive information over internet routes.

For the purpose, many researchers have designed efficient encryption algorithms considering the various cryptanalytic attacks which evolved as the security enhancements were formulated. One of the most popular methods was chaotic based image encryption[1] [2] scheme which was comparatively easier to implement with low running time complexity. Moreover the ciphers produces were extremely sensitive to the smallest deviation in the keys used in the algorithms. The slightest bit change in the key led to formation of ciphered image which was considerably different from the other key. Hence high sensitivity to keys was a major advantage for the chaotic based encryption schemes.

The chaotic image encryption scheme gained huge popularity out of its easy to implement technique. It used one-dimensional chaotic map which was later on evaluated to have high vulnerability to the latest cryptanalytic approaches.

Most of the encryption strategies proposed during those times were good at operating on the 2 dimensional gray scale images as well as binary images. But as the technology advancements led to huge evolution of cutting edge technology along with high resolution cameras hand held cameras which could capture images [3] with resolution as high as 20 mega pixel, the need to work upon the color images became the need of the hour. The various color models like RGB, YUV, CMY and YIQ [4] [5] color models were studied upon to check the feasibility of the available encryption algorithms on them. The existing algorithms which pertained during the time had been exploited and compromised by various attacks that application of them on these color models did not seem to be serving the purpose.

The chaotic based encryption was applied on the color images in order to encrypt them but since the potential of the chaotic based encryption has already been limited since it involved the shift of entire pixels as such across the rows and columns. The basic property of the pixel being shifted remained inherent in the pixel which helped the cryptanalyst to apply various probabilistic and Gaussian distribution theories to find the correlation analysis among the pixels and re order them to an extent to reveal some information out of the image.On the other hand researchers have developed other encryption methods [6] [7] [8]. Considering thisweaknessof the approach another algorithm was developed [9] in which the pixels were further split up into its triad components called the R G B values. These values denoted the intensity levels of the Red, Green and Blue shades denoted by values ranging from 0 – 255. The algorithm worked upon each of these R G and B values separately. All the three components were selected to work upon by the algorithm with 3 different keys for each of these components. This method decomposed each pixel in the image and scattered the native R G B components of that pixel into other pixels of the images hence making it totally difficult to reform the pixels which is being broken down. The basic property of the pixel was lost unless the exact three R G and B values were brought back to form that pixel again. This led to a considerable change in the color profile of the image since the pixels lost its basic intensity levels and grey scale intensity levels in the process. The key space was very large since the horizontal and vertical shifting of each R G and B components had its own set of keys. The ciphering was made more complex by choosing number of vertical and horizontal shift patters. Larger the shifts, more as the complexity of the ciphering process which ensured higher magnitude of security level at each change in shift pattern. But the simple linear approach was still vulnerable to the attack since if the slightest information of how the previous shift was performed can be guessed, the attacker can reverse execute the shift process and arrive at the less and easily analyzable form. The main weakness of this method was out of the inherent weakness of the algorithm which worked upon the whole image. The shifts were performed in a circular pattern across the dimension of the image. Suppose the dimension of the image was 100 by 100 pixels and the shift was done with the key value 20, the effect was same if the key used was 20, 120, 220 and so on. Since it was a circular shift, the addition or subtraction of multiples of 100 returned an alternate value which could be used in place of the original key.

2. PROPOSED METHOD

Considering the prime weakness in the above method, we propose a new technique to this problem in which instead of considering the whole image as one to work upon, we slice the images [10] into n number of parts and the above method is applied to each of these slices separately. Each size can be customized separately by choosing the number of rows and columns which in turn can be derived from the encryption keys. The secrecy factor introduced here provides a strong resistance against the multiple available alternatives for a single value of key as studied in the previous section.

To add to the security, the proposed method also induced the shuffling of the custom slices by inter changing the location of these slices from its actual image place location to different location repeating the shift algorithm with the new arrangement of the slices.

Changing the location of slices adds to the confusion for the cryptanalyst who is not provided with the knowledge of where and how the shuffling is performed and what is the updated location of each slice at that stage of the process.

Here we arbitrarily take the number of slices generated ‘n’ at each slicing process as 4 for the purpose of simulation and easier understandability in the algorithm. Once the slices are updated with new positions, a difference slice dimension can be applied to the current image making further split of the previous slices into further pieces. In this way, we can achieve considerable amount of security in the ciphered image by dynamically deciding at each stage of how many more slicing and shuffling should be done. Hence we can custom the security levels and maintain a good tradeoff between the security and the running time complexity of the encryption and decryption process.

3. ALGORITHM

Here we consider a plaintext image ‘e’which consists of M rows and N columns which is to be encrypted using the proposed algorithm.

In the initial step, we slice the image into 4 parts using the Slicing () function defined below. The variable xmas is initialized as M and ymax as N. The slicing function divides the entire image into 4 part and feeds each of this part into a PerformEncryption() function which acts upon each of these slices with its own calculated boundary limits calculated inside the Slicing() function. The variable quad is used to specify which quadrant is being used in the step.

Slicing (e, M, N)

```
1:   An Input Image img with x coordinates 1 to xmax
2:   Y coordinates 1 to ymax
3:   Initialize quad=1, e=img
4:   Loop while quad<=4
4.1        if quad=1
4.2        x1=1, x2=xmax/2, y1=1, y2=ymax/2
4.3        else if quad=2
4.4        x1=1, x2=xmax/2, y1=ymax/2+1, y2=ymax
4.5        else if quad=3
4.6        x1=xmax/2+1, x2=xmax, y1=1, y2=ymax/2
4.7        else
4.8        x1=xmax/2+1, x2=xmax, y1=ymax/2+1, y2=ymax
4.9        Invoke PerformEncryption(e,x1,x2,y1,y2,quad)
4.10       Increment quad by 1
4.11 EndLoop
5: Terminate
```

The PerformEncryption() method takes each of the quadrants with their coordinate limits and started performing the encryption process by deploying the shifting of R G B components among the pixels.[reference of old paper]. The PM[] array called asShift pattern mask array consists of binary digits 1’s and 0’s. The length of this array is the total number of vertical and horizontal shifts done in the encryption process. Each 1 triggers a circular vertical shift and a 0 triggers the invocation of circular horizontal shift. ThePM[ ] can be made a part of the key or else supplied separately. With the increase in the length of the mask, the security as well as running time for encryption process increases linearly.

PerformEncryption (e,x1,x2,y1,y2,quad)

```
1: Supply PM[ ] array
          1.1:      Initialize Counter=1, initJump= Any Arbitrary Integer
          1.2:      Loop while PM[counter] is not NULL
          1.2.1:if PM[counter]=0
                    Invoke HORIZONTAL_Shift(e,x1,x2,y1,y2, αr[counter], αg[counter], αb[counter],quad)
                    Increment counter by 1.
```

```
            Endif
        1.2.2: if PM[counter] = 1
            Invoke VERTICAL_Shift(e,x1,x2,y1,y2, αr[counter], αg[counter], αb[counter],quad)
            Increment counter by 1.
            Endif
            Endloop
2: Terminate
```

The PerformEncryption method uses another set of arrays namely αr[counter], αg[counter], αb[counter] which holds in it the different integers for R, G and B component shifts. This ensures that in each successive row, the displacement of a component doesn't remain a constant. Else it will result in the simple circular shift of the entire color component and hence it becomes a favorable condition for the cryptanalyst since guessing the shift of a single row is enough to know by how much are the other rows also shifted. The same entity is also used in the HORIZONTAL_Shift function also to provide a wider scattering of the R G B components from its native pixel position.

VERTICAL_Shift(e,x1,x2,y1,y2, α_r[counter], α_g[counter], α_b[counter],quad)

```
1: Input image with its coordinate limits x1 to x2,y1 to y2.
2: αr[counter], αg[counter], αb[counter]
3: ΔR= initJump + αr[counter]
4: ΔG= initJump+  αg[counter]
5: ΔB= initJump + αb[counter]
6: Loop and Repeat steps for ColC = x1 to ColC= x2
        Do Circular Vertical Shift of R values at ColCthcolumn by ΔR pixels
        Do Circular Vertical Shift of G values atColCth column by ΔG pixels
        Do Circular Vertical Shift of B values atColCth column by ΔB pixels
        ΔR = ΔR + αr[counter]
        ΔG = ΔG + αg[counter]
        ΔB = ΔB + αb[counter]
Endloop
7: Return
```

HORIZONTAL_Shift(e,x1,x2,y1,y2, α_r[counter], α_g[counter], α_b[counter],quad)

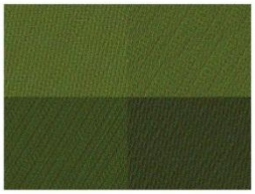
```
1: Input image with its coordinate limits x0 to xmaxy0 to ymax.
2: αr[counter], αg[counter], αb[counter]
3: ΔR= initJump + αr[counter]
4: ΔG= initJump+  αg[counter]
5: ΔB= initJump + αb[counter]
6: Loop and Repeat steps for RowC = y1 to RowC= y2
        Do Circular Horizontal Shift of R values at RowCth rowby ΔR pixels
        Do Circular Horizontal Shift of G values at RowCth row by ΔG pixels
        Do Circular Horizontal Shift of B values atRowCth row by ΔB pixels
        ΔR = ΔR + αr[counter]
        ΔG = ΔG + αg[counter]
        ΔB = ΔB + αb[counter]
Endloop
```

4. SIMULATION RESULTS

The simulation of the above algorithm was performed on the MATLAB Version 7.0.1 to verify the effectiveness of the proposed algorithm.The length of pattern mask PM[] was limited to 4 and the values [1,0,1,0] was stored in the mask. This invoked a vertical shift followed by a horizontal shift and once again preceded by another set of a vertical and a horizontal and vertical shift. The displacement factors were selected as 20, 25 and 30 for each of the pixel intensity components. The key values for R G and B value displacement were selected as 12, 24 and 36.

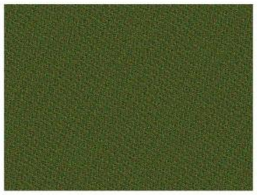
The MATLAB code for the algorithm was written and tested with the above values and the output is shown below.

4.1 Plane Image



4.2 Cipher Image after Slicing

4.3 Plane Image



4.4 Cipher Image afterShuffling slices

This MATLAB code was tested for 15 different size images having different resolution and tabulated outputsof the experiments is shown in table 4.1. Running time increases with the increase in image resolution and less dependent on the size of image. Its observed that as the resolution of image increasing, its execution time also increasing.

Sr. No.	Running Time in seconds	Resolution	Size in KB
1	1.937	320*240	46
2	11.5	800*600	28
3	11.594	800*600	70
4	12.891	800*600	104
5	16.265	1024*576	156
6	16.609	1024*576	357
7	29.735	1280*800	206
8	30.969	1024*1024	784
9	36.09	1440*900	133
10	59.172	1920*1080	4299
11	62.516	1920*1080	249
12	67.172	1920*1080	1573
13	98.797	2048*1536	522
14	102.094	2048*1536	639
15	259.235	3264*2448	3256

Table 4.1 Algorithm running time based on image size and resolution.

5. DISCUSSION AND CONCLUSION

The slicing and reshuffling of the image in steps between the processes has proved to be really effective in terms of the security analysis. The extra chaos induced into the image file after R G B component shifting has proved the increase of security of the image against all possible attacks available currently.

Our future research on this is focused on the random selection of the split points on the basis of which the image can be sliced. The slicing point can be made random as any point across the image by calculating it from the encryption keys.

6. REFERENCES

[1] RashidahKadir, Rosdiana Shahril2 and MohdAizainiMaarof, "A modified image encryption scheme based on 2D chaotic map" 978-1-4244-6235-3/10/$26.00 ©2010 IEEE.

[2] MazleenaSalleh, Subariah Ibrahim & Ismail FauziIsnin, "Image Encryption Algorithm Based On Chaotic Mapping".

[3] Abhishek Gupta, SandeepMahapatra and Karanveer Singh, "Data Hiding in Color Image Using Cryptography with Help of ASK Algorithm", 978-1-4577-0240-2/11/$26.00 ©2011 IEEE.

[4] Che-Yen Wen and Chun-Ming Chou, "Color Image Models and its Applications to Document Examination", Forensic Science Journal 2004;3:23-32.

[5] Information available via www at http://software.intel.com/sites/products/documentation/hpc/ipp/ippi/ippi_ch6/ch6_color_models.html.

[6] SaharMazloom and Amir-MasudEftekhari-Moghadam, "Color Image Cryptosystem using Chaotic Maps", 978-1-4244-9915-1/11/$26.00 ©2011 IEEE.

[7] R.E. Sawda, A.A.Falou, Gilles Keryer and A.Assoum, Image encryption and decryption by means of an optical phase mask. 0-7803-9521-2/2006 IEEE.

[8] Rami El Sawda, AymanAlfalou, HabibHamam, "RGB Colored Image Encryption Processes Using Several Colored Keys Images," fgcn, vol. 2, pp.594-598, Future Generation Communication and Networking (FGCN 2007) - Volume 1, 2007.

[9] Reji Mathews, Amnesh Goel, PrachurSaxena&VedPrakash Mishra, "Image Encryption Based on Explosive Inter-pixel Displacement of the RGB Attributes of a PIXEL", Proceedings of the World Congress on Engineering and Computer Science 2011 Vol I WCECS 2011, October 19-21, 2011, San Francisco, USA. ISBN: 978-988-18210-9-6.

[10] Ping Xu, Jianjun Zhao and Dihua Wang, "A selective image encryption algorithm based on hyper-chaos", 978-1-61284-486-2/11/$26.00 ©2011 IEEE.