http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 9 (3): 446-452, 2010
ISSN 1812-5638

Color Image Encryption Based on Secret Sharing and Iterations

Hao Luo, Fa-Xin Yu, Hui Li and Zheng-Liang Huang
School of Aeronautics and Astronautics, Zhejiang University,
Hangzhou, 310029, Peoples's Republic of China

Abstract: This study proposes a novel simple scheme for color image encryption. The RGB color components of the input image are permuted and transformed into the YCbCr color space. Then three simultaneous equations are constructed for secret sharing of the luminance and two chrominance components. After several iterations, the output of this equation set is the encrypted content. These encrypted channels are recomposed to the encrypted image. In image decryption, we merely need to solve the simultaneous equation set based on Largrange's interpolation with several inverse iterations. The decrypted image is of very high quality for only slight degradation is introduced due to the color space mapping. Security of our scheme is well preserved as long as the permutation key kept secret. Experimental results validate the effectiveness of the proposed method.

Key words: Color image encryption, secret sharing, Largrange's interpolation, YCbCr color space

INTRODUCTION

Digital image encryption plays an important role in multimedia security due to the development of Internet and the increasing requirement for image transmission. Nowadays many schemes have been developed for image encryption. Most of them are designed for gray level image encryption based on various transforms such as Hartley transform (Li and Zhao, 2009; Ahmad and Liu, 2009; Singh and Sinha, 2009) fractional wavelet transform (Chen and Zhao, 2008) fractional Fourier transform (Jin and Yan, 2007; Liu *et al*., 2009a, b; Liu and Liu, 2007; Joshi *et al*., 2009) and so on. Besides transform domain based methods, some spatial domain based image encryption schemes have been reported in literatures. A pixel-based scrambling scheme is proposed for gray level medical image encryption in (Hu and Han, 2009). In this system, the input image is decomposed into several binary bitplanes and then the simple exclusive-OR (XOR) operation is performed in an innovative way.

This letter proposes a novel simple scheme for color image encryption. The principle of image secret sharing combined with iterations is exploited in our spatial domain based method. The XOR-based method (Hu and Han, 2009) can be essentially regarded as a module-2 mechanism. In contrast, a module-p mechanism is employed in our scheme, where p is a specific prime integer. In image encryption, the pixel values are directly input for secret sharing after a pixel-wise color space mapping. As once secret sharing is not enough to scramble the pixel values randomly, several iterations of secret sharing are executed. The image decryption is a task to solve an equation set consisting of three simultaneous equations with Largrange's interpolation.

Our color image encryption method is motivated from the (r, n)-threshold scheme proposed by Shamir (1979). Suppose, we encrypt a secret integer s into n shares held by n participants. In the (r, n)-threshold paradigm, two properties must be maintained. One is that any r or more than r shares can be used to recover the secret integer and the other is that any r-1 or fewer shares cannot reconstruct it. In particular, the sharing procedures are described below.

To encrypt s into n shares, a prime p is randomly selected and an r-1 degree polynomial is constructed as:

$$q(x) = (s+a_1x+...+a_{r-1}x^{r-1}) \bmod p \quad (1)$$

where, the coefficients a_i (i = 1, 2, …, r-1) is a randomly chosen positive integer. Both s and a_i are smaller than p. Accordingly, the set consisting of n shares q(j) (j = 1, 2, …, n) can be generated as:

$$\begin{aligned} &q(1) = (s + a_1 + ... + a_{r-1}) \bmod p \\ &q(2) = (s + 2\times a_1 + ... + 2^{r-1}\times a_{r-1}) \bmod p \\ &... \\ &q(n) = (s + n\times a_1 + ... + n^{r-1}\times a_{r-1}) \bmod p \end{aligned} \quad (2)$$

According to Largrange's interpolation, the secret data s can be recovered by solving an equation set

Corresponding Author: Fa-Xin Yu, School of Aeronautics and Astronautics, Zhejiang University, Hangzhou, 310029, Peoples's Republic of China

constructed by any r shares. For example, suppose s = 3 is shared with a (2, 4)-threshold model and p is set as 7. Then an equation is defined as q(x) = (3+5x)mod7, where, the coefficient 5 is randomly selected. Thus, four shares q(1) = 1, q(2) = 6, q(3) = 4 and q(4) = 2 are computed. If q(2) and q(3) are collected, we have q(2) = 6 = (s+2a_1)mod7 and q(3) = 4 = (s+3a_1)mod7. In this way, the secrets can be solved as 3 based on these two equations.

In this basic (r, n)-threshold scheme, we can find that each share is of the same size as that of the input secret and in practice the coefficient is also calculated (e.g., a_1 = 5 in the above example) along with the secret integer during decryption. In order to reduce the share size, (Thien and Lin, 2002) extended the (r, n)-threshold model to share an 8-bit gray level image. Their idea is to employ all coefficients of the r-1 degree polynomial in Eq. 1 to carry the input image data. Specifically, the secret image is partitioned into a set of non-overlapping segments, each having r pixels. For each segment j, an r-1 degree polynomial is defined as:

$$q_j(x) = (p_0 + p_1x + ... + p_{r-1}x^{r-1}) \bmod p \quad (3)$$

where, p_0, p_1,…, p_{r-1} are the r pixel values of the segment. Then $q_j(1)$, $q_j(2)$, …, $q_j(n)$ can be computed and sequentially assigned to n share holders. As each share receives one of the generated pixels, it is 1/r size of the secret image. Thus transmission is speeded up and the storage space is saved.

In Thien and Lin's method, p is set as 251 for the largest prime between 0 and 255 is 251. Hence, the input pixel values must be truncated into 0 to 251 before sharing. Obviously, this truncation is a lossy process. In other words, some original information is lost when encrypting a pixel with the value larger than 251. Thien and Lin (2002) also tailored a lossless sharing scheme for this special case. That is, some extra space is used to record the difference between 250 and 255. Consequently, the size of each share is more or less larger than 1/r of the input image.

PROPOSED METHOD

Our goal is to encrypt a 24-bit color image with the red (R), green (G) and blue (B) color components of each pixel P(u, v) recorded by 8 bits, respectively. In our context, P(u, v) denotes the pixel in the u-th row and v-th column of the image. The image encryption and decryption procedures are described below.

Color image encryption: Suppose the original image I is of size M×N, then the encryption steps can be given as follows.

Step 1. Image permutation: This step is to decorrelate the color component correlation of the input image. Generate a pseudo random number sequence with M×N×3 elements according to a key K. Suppose the RGB components of P(u, v) are represented as P_r(u, v), P_g(u, v) and P_b(u, v), respectively. Concatenate the RGB planes of P(u, v) into a sequence S with the raster scanning order as:

$$\begin{aligned} S = [&P_r(1,1), P_r(1,2), ..., P_r(M,N), P_g(1,1), P_g(1,2), ..., \\ &P_g(M,N), P_b(1,1), P_b(1,2), ..., P_b(M,N)] \end{aligned} \quad (4)$$

Then permute S into S_p with the pseudo random number sequence as:

$$\begin{aligned} S_p = [&P_{pr}(1,1), P_{pr}(1,2), ..., P_{pr}(M,N), P_{pg}(1,1), P_{pg}(1,2), ..., \\ &P_{pg}(M,N), P_{pb}(1,1), P_{pb}(1,2), ..., P_{pb}(M,N)] \end{aligned} \quad (5)$$

Obviously, S_p can be rearranged into a permuted color image PI. That is, P_{pr}(1, 1), P_{pr}(1, 2),…, P_{pr}(M,N) are rewritten to the red channel of PI, P_{pg}(1,1), P_{pg}(1,2),…, P_{pg}(M,N) are rewritten to the green channel of PI and P_{pb}(1,1), P_{pb}(1, 2),…, P_{pb}(M,N) are rewritten to the blue channel of PI. Actually, each RGB component of the permuted pixel is retrieved from that of some other pixel of I.

Step 2. RGB to YCbCr color space transform: In this step, the permuted image PI is transformed from RGB into the YCbCr color space image PI'. The YCbCr system is a popular luminance-chrominance space. As the Human Visual System (HVS) is more sensitive to changes in luminance than those in chrominance, the main advantage of luminance-chrominance systems (e.g., YCbCr, YUV, YIQ) is that some of the chrominance information can be discarded without causing noticeable artifacts. In our case, the RGB to YCbCr space mapping is defined as follows:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \frac{1}{255} \times \begin{bmatrix} 65.481 & 128.553 & 24.966 \\ -37.797 & -74.203 & 112 \\ 112 & -93.786 & -18.214 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \quad (6)$$

where, Y stands for the luminance component, Cb and Cr represent the blue and red chrominance components. Obviously, the RGB to YCbCr space transform is a lossy

process. Each color component in the RGB space lies in the range of 0 to 255, while in the YCbCr space, the Y component is transformed into the range of 16 to 235 and Cb and Cr are both scaled to the range of 16 to 240. In general, the visual distortion introduced by the RGB to YCbCr color space transform is very slight such that it cannot be distinguished by human eyes.

Actually, the color space transform is essential because the largest prime can be used between 0 and 255 is 251 and the color component of each pixel in the YCbCr space is smaller than 251. The color channel decomposition and transform of the 512×512 Lena image are shown in Fig. 1.

Step 3. Secret Sharing: This step is to encrypt the YCbCr space image PI'. The encryption operation is given as Eq. 7. Suppose the Y, Cb and Cr components of the encrypted image EI^1 are EI^1_y, EI^1_{cb} and EI^1_{cr}, respectively. Here the superscript 1 denotes they are results of the first iteration of secret sharing.

$$\begin{aligned} EI^1_y(u,v) &= (PI'_y(u,v) + PI'_{cb}(u,v) + PI'_{cr}(u,v)) \bmod 251 \\ EI^1_{cb}(u,v) &= (PI'_y(u,v) + 2\times PI'_{cb}(u,v) + 4\times PI'_{cr}(u,v)) \bmod 251 \\ EI^1_{cr}(u,v) &= (PI'_y(u,v) + 3\times PI'_{cb}(u,v) + 9\times PI'_{cr}(u,v)) \bmod 251 \end{aligned} \quad (7)$$

Repeated the above operations on all pixels of PI' and then EI^1 can be obtained by EI^1_y, EI^1_{cb} and EI^1_{cr}, recomposition.

Step 4. Iterations: Although, the secret sharing in step 3 converts the PI' into a random noise-like image EI^1, this single procedure cannot scramble the pixel values evenly distributed. In other words, three histograms of EI^1 in the R, G and B components are not flat. Therefore, the operations in the Step 3 are iterated as Eq. 8.

$$\begin{aligned} EI^{k+1}_y(u,v) &= (EI^k_y(u,v) + EI^k_{cb}(u,v) + EI^k_{cr}(u,v)) \bmod 251 \\ EI^{k+1}_{cb}(u,v) &= (EI^k_y(u,v) + 2\times EI^k_{cb}(u,v) + 4\times EI^k_{cr}(u,v)) \bmod 251 \\ EI^{k+1}_{cr}(u,v) &= (EI^k_y(u,v) + 3\times EI^k_{cb}(u,v) + 9\times EI^k_{cr}(u,v)) \bmod 251 \end{aligned} \quad (8)$$

where, k is a positive integer. According to Eq. 8, we can obtain the scrambled image after k+1 times of iterations EI^{k+1} from the version of EI^k.

Color image decryption: The image decryption is the inverse process of the encryption, which can be illustrated below:

Step 1. Simultaneous equation set construction and solution: The first step is to construct the simultaneous equation set as Eq. 8. It can be further rewritten as Eq. 9.

$$\begin{bmatrix} EI^{k+1}_y(u,v) \\ EI^{k+1}_{cb}(u,v) \\ EI^{k+1}_{cr}(u,v) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \end{bmatrix} \times \begin{bmatrix} EI^k_y(u,v) \\ EI^k_{cb}(u,v) \\ EI^k_{cr}(u,v) \end{bmatrix} \bmod 251 \quad (9)$$

It can be expressed as:

$$\begin{bmatrix} EI^k_y(u,v) \\ EI^k_{cb}(u,v) \\ EI^k_{cr}(u,v) \end{bmatrix} \bmod 251 = \begin{bmatrix} 3 & -3 & 1 \\ -2.5 & 4 & -1.5 \\ 0.5 & -1 & 0.5 \end{bmatrix} \times \begin{bmatrix} EI^{k+1}_y(u,v) \\ EI^{k+1}_{cb}(u,v) \\ EI^{k+1}_{cr}(u,v) \end{bmatrix} \quad (10)$$

Fig. 1: Color channel decomposition of the 512×512 image Lena: the original image in the RGB space and the R, G and B channel planes (the top row, from left to right); the transformed image in the YCbCr space and the Y, Cb and Cr channel planes (the bottom row, from left to right)

where, $\begin{bmatrix} 3 & -3 & 1 \\ -2.5 & 4 & -1.5 \\ 0.5 & -1 & 0.5 \end{bmatrix}$ is the inverse matrix of $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \end{bmatrix}$.

Thus, the $\begin{bmatrix} EI_y^k(u,v) \\ EI_{cb}^k(u,v) \\ EI_{cr}^k(u,v) \end{bmatrix}$ can be obtained by solving the equation set as Eq. 10. That is, EI^k is recovered.

Step 2. Inverse iterations: As the iterations in Eq. 7 and 8 are reversible in nature, we can obtain the EI^1 with inverse iterations. It is a repeated process of tracing the source EI^k of EI^{k+1} by solving an equation set until k = 1. In this way, the permuted image PI' can be finally recovered according to EI^1.

Step 3. YCbCr to RGB color space transform: The permuted image PI' in the YCbCr space is transformed into the RGB space image PI with the inverse mapping given in Eq. 6.

Step 4. Image inverse permutation: The last step is to apply the permuted key K in encryption to inversely permute PI and thus the decrypted image is obtained.

RESULTS AND DISCUSSION

Here, a series of experiments are conducted to demonstrate the validity of the proposed scheme. The metric Peak Signal to Noise Ratio (PSNR) is used to quantitatively evaluate the quality of the decrypted image compared with its original version. For a 24 bit color image, the PSNR is defined as:

$$PSNR = 10 \times \log_{10} \frac{255^2}{MSE} (dB) \quad (11)$$

where, MSE is the mean square error defined as:

$$MSE = \frac{1}{M \times N \times 3} \sum_{c \in (R,G,B)} \sum_{u=1}^{M} \sum_{v=1}^{N} (I_c(u,v) - I'_c(u,v))^2 \quad (12)$$

where, I and I' denote the original image and the decrypted image of size M×N respectively. Generally, a higher PSNR value indicates better quality of the decrypted image.

Six 512×512 color images are selected to evaluate the average performance of our scheme. As shown in Fig. 2, the test images of Lena, Splash, Baboon, Airplane, Lake and Peppers are selected from the University of Southern California image database in Miscellaneous with the corresponding filenames 4.2.04.tiff, 4.1.01.tiff, 4.1.03.tiff, 4.1.05.tiff, 4.1.06.tiff, 4.1.07.tiff, respectively.

The PSNR results of the six test images, Lena, Splash, Baboon, Airplane, Lake, Peppers, are 52.10, 52.17, 52.11, 52.10, 52.12 and 52.19dB, respectively (as shown in Table 1). It is easy to find that all of the PNSR values of

Table 1: PSNR values of the decrypted images

Test image	PSNR(dB)
Lena	52.10
Splash	52.17
Baboon	52.11
Airplane	52.10
Lake	52.12
Peppers	52.19

Fig. 2: Six 512×512 test images of Lena, Splash, Baboon, Airplane, Lake, Peppers (from left to right and top to bottom)

Fig. 3: Experimental results on the Lena image, (a) the original image, (b) the permuted image in the RGB color space, (c) the permuted image in the YcbCr color space, (d) the encrypted image with three times of iterations, (e) the decrypted image in the RGB color space before inverse permutation and (f) the decrypted image with PSNR = 100.23 dB

these decrypted images are larger than 52 dB. In general, human eyes cannot distinguish the distortions when the image quality is larger than 30 dB. Therefore, all the decrypted images qualities are acceptable in many applications.

We also evaluate our scheme's performance on the Lena Image. The permuted image in the RGB space and its YCbCr space version are shown in Fig. 3b and c, respectively. The encrypted image with three times of iterations (i.e., k = 2) as shown in Fig. 3d looks like random noise. That is, no meaningful information can be revealed from it. During image decryption, the decrypted content is transformed backward into the RGB color space first as shown in Fig. 3e and then to the finally decrypted image as shown in Fig. 3f via inverse permutation. The image quality degradation is slightly introduced during the RGB to YCbCr color space transform. All the other operations in our scheme do not lead to any degradation, i.e., they are lossless. Fortunately, as the PSNR value of the decrypted Lena image (Fig. 3f) is 52.10 dB, we can find that the color space mapping does not introduce severe distortions to the decrypted image.

The iteration parameter k in our scheme is quite essential. Obviously, a larger k means more computation complexity is involved. A large quantity of experimental results on diverse color images show that three times of iterations (i.e., k = 2) are quite enough to scramble the pixel values evenly distributed.

The airplane image is adopted as the test image to show the functions of the iteration parameter k. The encrypted images as shown in Fig. 4b-d are obtained with once, twice and three times iterations, respectively. The decrypted images corresponding to Fig. 4b-d are shown in Fig. 4e-g. All of their PSNR values are equal to 52.10 dB, for the image quality degradation is caused by the same RGB to YCbCr space transform. The RGB channel histograms of these encrypted images are shown in Fig. 4h-j, respectively. It is easy to find that the RGB channel histograms of Fig. 4b are fluctuant and those of Fig. 4c are relatively flat. In contrast, the histograms of the Fig. 4c are flat in all bins.

The security of our scheme is guaranted by the permutation key K. As our permutation is performed on the concatenated R, G and B components, the possibility of guessing the right permutation to an M×N image is only

$$\frac{1}{(M\times N\times 3)!}\ .$$

For example, to an image of the size 4×4 pixels, this possibility is approximate to 8.06×10^{-62}. To a 512×512 image, this possibility is an extremely small number, i.e.,

$$\frac{1}{(512\times 512\times 3)!}\ .$$

Therefore, the security of our scheme can be well preserved by keeping the permutation key secret.

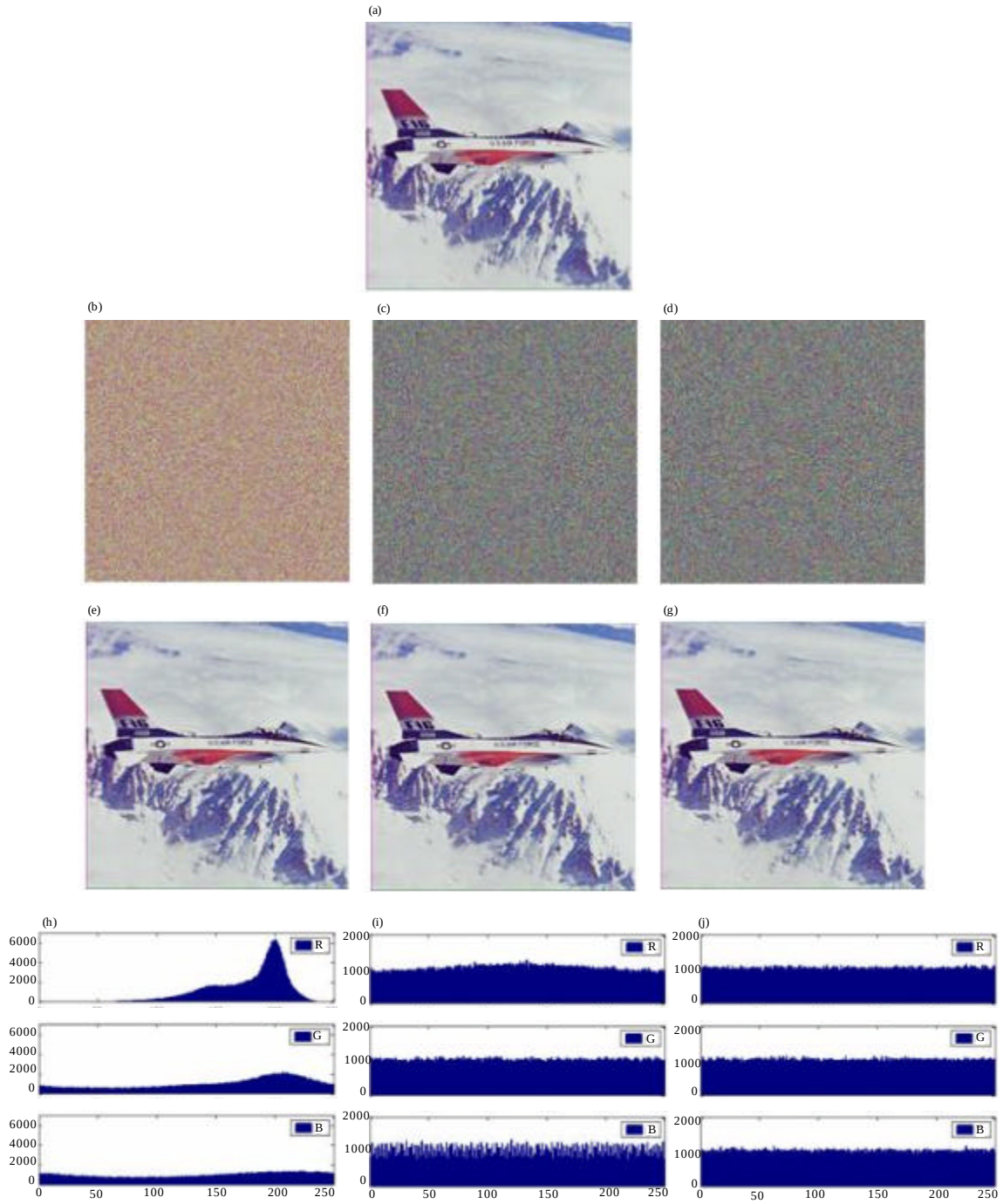

Fig. 4: Experimental results on the Airplane image, (a) the original image, (b-d) the encrypted images with once, twice and three times of iterations, (e-g) the decrypted images corresponding to (b-d), (h-j) the RGB channel histograms of (b-d)

CONCLUSIONS

A novel color image encryption scheme is proposed in this letter. It is based on image pixels secret sharing and iterations. Although, some slight image degradation is introduced in the color space mapping, the average PSNR values of the decrypted images can be maintained larger than 100 dB. Our scheme is useful in many applications where, a high quality decrypted image is required.

ACKNOWLEDGMENT

The authors would like to give many thanks to Dr. Hua Chen, Zhejiang University, for the security discussions of the proposed method.

REFERENCES

Ahmad, M.A. and S. Liu, 2009. Image encryption based on double random amplitude coding in random Hartley transform domain. Optik Int. J. Light Electron. Opt., 10.1016/j.ijleo.2008.12.006

Chen, L. and D. Zhao, 2008. Image encryption with fractional wavelet packet method. Optik Int. J. Light Electron. Opt., 119: 286-291.

Hu, J. and F. Han, 2009. A pixel-based scrambling scheme for digital medical images protection. J. Network Comput. Appl., 32: 788-794.

Jin, W. and C. Yan, 2007. Optical image encryption based on multichannel fractional Fourier transform and double random phase encoding technique. Optik Int. J. Light Electron. Opt., 118: 38-41.

Joshi, M., C. Shakher and K. Singh, 2009. Logarithms-based RGB image encryption in the fractional Fourier domain: A non-linear approach. Opt. Laser Eng., 47: 721-727.

Li, X. and D. Zhao, 2009. Optical color image encryption with redefined fractional Hartley transform. Int. J. Light Electron. Opt., 10.1016/j.ijleo.2008.10.008

Liu, Z. and S. Liu, 2007. Double image encryption based on iterative fractional Fourier transform. Opt. Commun., 275: 324-329.

Liu, Z., J. Dai, X. Sun and S. Liu, 2009a. Triple image encryption scheme in fractional Fourier transform domains. Opt. Commun., 282: 518-522.

Liu, Z., Q. Li, J. Dai, X. Sun, S. Liu and M.A. Ahmad, 2009b. A new kind of double image encryption by using a cutting spectrum in the 1-D fractional Fourier transform domains. Opt. Commun., 282: 1536-1540.

Shamir, A., 1979. How to share a secret. Commun. ACM, 22: 612-613.

Singh, N. and A. Sinha, 2009. Optical image encryption using improper Hartley transforms and chaos. Optik Int. J. Light Electron. Opt., 10.1016/j.ijleo.2008.09.049.

Thien, C.C. and J.C. Lin, 2002. Secret image sharing. Comput. Graphics, 26: 765-770.