Journal of Engineering and Applied Sciences 13 (8): 2038-2041, 2018
ISSN: 1816-949X

Color Image Encryption with a Key Generated by Using Magic Square

[1]Ahmeed Suliman Farhan, [1]Sudad H. Abed and [2]Fouad H. Awad
[1]Computer Center, University of Anbar, Ramadi, Iraq
[2]College of Business Informatics,
University of Information Technology and Communications, Ramadi, Iraq

Abstract: This study presents a new method of encrypting color digital images using magic square. The proposed method generates an encryption key by using magic square and encrypts images using the generated key. The encryption method changes the RGB (Red, Green and Blue) values of each pixel of the plain-text image using the generated key. The process starts by splitting the plain-text image into three arrays that hold RGB values. After encrypting each segment of 256 pixels of the RGB arrays, the encrypting key is permuted by using magic square. At the end, the encrypted image is generated by merging all three encrypted RGB arrays together. For the decryption of the cipher-image, the same process of encryption used but with the encrypted images as the input.

Key words: Image encryption, magic square, decryption, permutation, RGB values, pixel

INTRODUCTION

The digital images represent a popular type of multimedia and have been widely used in different areas. With the huge number of images that are transferring publicly in deferent sectors such as military, medical and political, there are always needs for effective methods to secure the data that images hold (Sowmiya *et al*., 2017). Since, digital images sizes are normally very large it is inefficient and expensive to encrypt them using traditional ciphers such as Data Encryption Standard (DES), Advanced Encryption Standard (AES) and the algorithm developed by Rivest, Shamir and Adleman (RSA), that are usually used for text encryption (Ye and Zhao, 2012). Image encryption can be mainly divided to two main ways which are image encryption by randomizing the pixel position and image encryption by changing the pixels values (Gupta *et al*., 2014). The proposed encryption algorithm changes the value of RGB values in each pixel of the image to generate the cipher-text image.

Magic square: Magic square is an array of two-dimension that can be given as n×n. The magic square has properties that make it interesting to study and use. For any given n×n magic squares, the summation of a row, columns or diagonal gives the same result which is called magic sum (Beck and Robins, 2015). The process of generating a magic square is quite sensitive where swapping the values between two cells will affects and change the configuration of others cells in the square (Pickover, 2002). Thus, using large deamination of magic squares in encryption will increase the complexity of the possibilities.

Literature review: In recent years, researchers started exploring using magic square in encryption for the properties that magic squares have. Sowmiya and his colleagues (Sowmiya *et al*., 2017) used magic squares to encrypt images by permuting the RGB values. Morover, Zhong *et al*. (2016) tried in their research to enhance the effect of encrypting images by pixel position permutating using magic squares. Our proposed method uses magic to encrypt images by changing the RGB values of each pixel.

MATERIALS AND METHODS

The main challenge of the research is to overcome the weak encryption key problem. The proposed method generates an encryption key by using the magic square. Then, encrypt each pixel by changing the RGB values by XOR operation between the generated key and the RGB values of each pixel. The proposed encryption method uses the same generated key for the encryption and decryption process.

Generating the key: In the encryption phase, generating the key starts with generating a magic square after determining three variables which are the size of magic square, start and difference. The size of the magic squares represents the dimensions of the square while the start variable represents the first value will be placed in the magic square where adding to it the third variable the difference value will generate the second value will be placed in the magic square. There are three type of magic square that can be generated differently from each other.

Corresponding Author: Ahmeed Suliman Farhan, Computer Center, University of Anbar, Ramadi, Iraq

Odd order magic square: Where the order n is of the form 2m+1 where m can be any positive integer. The example of this type includes the De la Loubere's method. The magic square dimensions will be 3×3, 5×5 and 7×7 and so on Rahma *et al*. (2016). The odd order magic square is constructed when it is needed in the proposed encryption algorithm to generate the encryption key using Siamese method which is also called de la Louberes (Prasartkaew and Choomchuay, 2012).

Doubly even order magic square: Where the order n is of the form 4 m where m can be any positive integer. The example of this type includes the Albrecht Durer's method. The magic squares dimensions will be 4×4, 8×8 and 12×12 and so on (Rahma *et al*., 2016). Agrippa: diagonal method (Prasartkaew and Choomchuay, 2012) was used to build the doubly even order magic square.

Singly even order magic square: Where the order n is of the form 2 (2m+1) were m scan be any positive integer. The example of this type includes the Philippe de la Hire's method. The magic squares dimensions will be 6×6, 10×10 and 14×14 and so on Rahma *et al*. (2016). Strachney method (Duan *et al*., 2015) was used to build singly even order magic square to generate the encryption key to be used in the proposed encryption method. After generating the mage square, a key K of length 256 digit is created with sequential values from 0-255 the default key before permutation.

0	1	2	3	...	253	254	255

The next step is to use the generated magic square to permutate the key K. The generated magic square is converted into one-dimension array called the permutation array p-array. The process of permutation the key is completed by looping on p-array and permutation the key K as stated in the function shown in Algorithm 1:

Algorithm 1; The permutation function:

```
public int[] create-permutation-array(int[] s, double[] p-array)
    {
      int i = 1, j = 1
      for (int x = 0; x < 256; x++)
      {
        for (int f = 0; f<p-array.Length; f++)
        {
          i = (int)((i + p-array [f]) % 256)
          j = (int)((s[f % 256]+p-array [i % p-array.Length]) % 256)
          int temp = K[i]
          K[i] = K[j]
          K[j] = temp
        } }
      return K
    }
```

Encryption method: The first step to encrypt a color image in the proposed encryption method is splitting the RGB values into three individual two-dimension arrays. The dimensions of each of the RGB arrays are as same as the plain-text image dimensions. The encryption key K is used to encrypt a segment of the image (256 pixels) by implementing XOR operation between the RGB arrays and the key K. After encrypting every segment of the image, the key K is permuted using the generated permutation array p-array to increase randomization.

The encryption process that is shown in Fig. 1 is applied on the three RGB arrays that hold the plain-tex image data. At the end of encrypting all the three RGB arrays, the results are merged together to form the cipher-text image. The decryption process to retrieve the original image is applied in the same way of encryption by using the same encryption key with XOR operation on the cipher-text image.

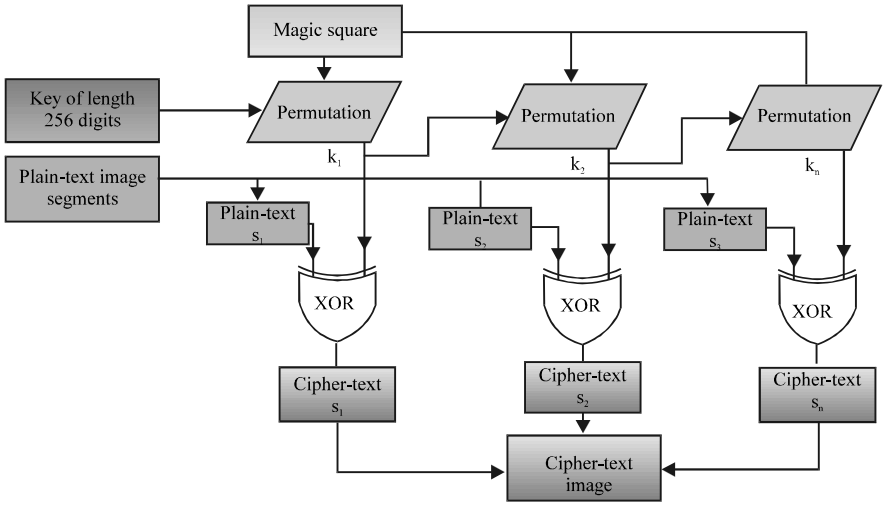

Fig. 1: Encryption process

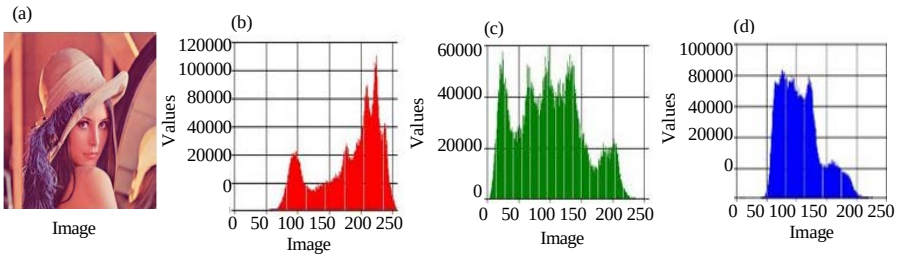

Fig. 2: The plan-text image and the histogram of RGB components

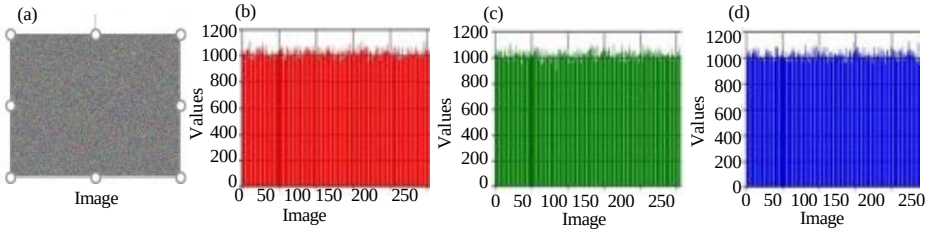

Fig. 3: The cipher-text image and the histogram of RGB components

Fig. 4: The image retrieved by decryption

RESULTS AND DISCUSSION

Experiment analysis: Several digital color images were encrypted and decrypted using the proposed encryption method. One of the used color image is Lena image of size 512×512 which is shown in Fig. 2 with the histogram of RGB components. The cipher-text image after applying the proposed encryption method with the histogram of RGB is show in Fig. 3.

It is clear from the histogram of RGB component in Fig. 2 and 3 how the pixels values have been changed and distributed on a wide range of the image. It is also observed the that the color distribution in the original image is significantly different from the distribution of the final encrypted image. The decryption has been implemented on the encrypted image in Fig. 4 with the same generated series of keys K and the result of the retrieved image can be shown in Fig. 4.

The strength of the proposed method lies in generating of ciphering key which keep secret and influenced by avalanche effect factor where changing one element in the key generating process can affect significantly on the final result of encryption and decryption process.

CONCLUSION

Many algorithms for encrypting images have been used for decades and researchers have started in recent years exploring methods to use magic squares in image encryption. The aim of this study is generate a strong encryption key and use it to encrypt digital color images using a proposed symmetric cipher.

The results of encrypting and decrypting images using the proposed encrypting method shows that cipher-text image has significant distribution of RGB components compared with the plain-text image regardless the encrypting process when the size of plain-text is very large.

REFERENCES

Beck, M. and S. Robins, 2015. Magic Squares. In: Computing the Continuous Discretely, Beck, M. and S. Robins (Eds.). Springer, New York, USA., ISBN:978-1-4939-2968-9, pp: 113-129.

Duan, Z., J. Liu, J. Li and C. Tian, 2015. Improved even order magic square construction algorithms and their applications in multi-user shared electronic accounts. Theor. Comput. Sci., 607: 391-410.

Gupta, A., N. Tiwari, M. Chawla and M. Shandilya, 2014. An image encryption using block based transformation and bit rotation technique. Intl. J. Comput. Appl., 98: 30-32.

Pickover, C.A., 2002. The Zen of Magic Square, Circles and Stars: An Exhibition of Surprising Structures Across Dimensions. Princeton University Press, Princeton, New Jersey, USA., Pages: 417.

Prasartkaew, C. and S. Choomchuay, 2012. Parity check matrix construction via Magic Square based Algorithm. Proceedings of the 2012 International Symposium on Communications and Information Technologies (ISCIT'12), October 2-5, 2012, IEEE, Queensland, Australia, ISBN:978-1-4673-1156-4, pp: 54-59.

Rahma, M.S., M.J.A. Hossen and O.A. Dawood, 2016. Public key cipher with signature based on diffie-hellman and the magic square problem. Eng. Tech. J., 34: 1-15.

Sowmiya, S., I.M. Tresa and A.P. Chakkaravarthy, 2017. Pixel based image encryption using magic square. Proceedings of the 2017 International Conference on Algorithms, Methodology, Models and Applications in Emerging Technologies (ICAMMAET'17), February 16-18, 2017, IEEE, Chennai, India, ISBN:978-1-5090-3379-9, pp: 1-4.

Ye, R. and H. Zhao, 2012. An efficient chaos-based image encryption scheme using affine modular maps. Int. J. Comput. Netw. Inform. Secur., 4: 41-50.

Zhong, W., Y.H. Deng and K.T. Fang, 2016. Image encryption by using magic squares. Proceedings of the 2016 International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI'16), October 15-17, 2016, IEEE, Datong, China, ISBN:978-1-5090-3711-7, pp: 771-775.