# A NOVEL IMAGE ENCRYPTION ALGORITHM BASED ON THE EXTRACTED MAP OF OVERLAPPING PATHS FROM THE SECRET KEY 

Mohammed A. Fadhil Al-Husainy ${ }^{1}$


#### Abstract

In the digital era, most types of transmitted data over the Internet are images. These images may carry secret information. So, it became necessary to have an effective encryption system to hide information inside these images. The use of random and large-sized key represents the main factors that give the encryption system the required strength and give encrypted data more protection. This paper presents a non-traditional encryption algorithm that uses a large key size ( 256 bytes) $=$ 2048 bits, to achieve a high degree of protection for the image. First, the algorithm represents the source image as a set of two-dimensional blocks of size $16 \times 16$ bytes and the secret key as a two-dimensional block of size $16 \times 16$ bytes. Second, transposition and substitution operations are implemented within each block of the image based on the extracted map of overlapped paths from the secret key block. Successive different keys are generated by using another way of implementation of the substitution and transposition operations in the secret key block. The recorded performance results, from the evaluation tests, have been compared with some well-known encryption algorithms and they proved that the proposed algorithm can be used effectively to protect images.


Mathematics Subject Classification. 68U10.

## 1. Introduction

In the world of digital information, privacy and confidentiality of information became one of the most important issues that force IT experts to develop innovative methods to protect digital information. In order to protect confidential information against unwanted users or illegal reproduction and amendments, there is a need to use different types of encryption methods.

Cryptosystems in general consists of two phases: encryption phase that transforms the original secret data to the encrypted data and decryption phase that transforms back the encrypted data to the original secret data. Usually, the encryption methods used two types of main operations to achieve data security: substitution operation that is working to replace the data values with new values; and transposition operation that is working to change the location of the data values to a new location. How to implement these operations are mainly based on the encryption method and the keys used in the method.

Keys play a major role in the strength of the encryption algorithm and achieve a higher degree of ambiguity about how to implement the above-mentioned operations. Therefore, these keys must be complex and large enough to achieve the highest possible protection of secret data.

[^0]Digital images are considered one of the means that have become widespread and exchange across different information networks. Most of these images often include information on a high degree of confidentiality. This urged many researchers to devise better ways of encryption to protect those images that contain confidential information.

Researchers in [1] presented a new method for image encryption by integrated pixel scrambling plus diffusion technique [IISPD]. The algorithm makes use of the full chaotic property of logistic map and reduces time complexity. The algorithm calculates the permutation address for row and column by xor'ing bits of the adjacent pixel values of the original image. The security analysis and its experimental analysis show that the proposed technique is highly sensitive to initial conditions. It also has higher key-space and a higher degree of scrambling.

Wadi and Zainal [2] made some modifications to enhance the performance of Advanced Encryption Standard (AES) algorithm in terms of time ciphering and pattern appearance especially when the AES use for ciphering the HD images. These modifications have been done by decreasing the number of rounds to one and replace the S-box with new S-box to decrease the hardware requirements.

Chaumonta et al. [3] presented an approach based on a color reordering algorithm after a quantization step. Based on a layer scanning algorithm, the color reordering generates gray level images and makes it possible to embed the color palette into the gray level images using a data hiding algorithm.

Different techniques were proposed by Seth et al. [4] which uses a selective block encryption technique to achieve the different goals of security such as Availability, Confidentiality, and Integrity.

Al-Husainy [5] produced good diffusion and confusion effects in the encrypted image through doing XOR and rotation bit operations on the image data.

Kuppusamy and Thamodaran [6] suggested an optimized hybrid image security mechanism for authentication and secrecy of images by means of Particle Swarm Optimization (PSO) in daubechies4 transform is illustrated in their paper. The issues such as authentication, robustness, security and statistical attacks have been solved by the suggested mechanism. To form the image hash and select high energy coefficients for partial encryption, the PSO technique is employed by selecting feature vectors of the image.

Sivakumar and Venkatesan [7] used Matrix Reordering (MR) by employing kind of scanning and simple XOR operation to produce a novel approach for encrypting digital images. The MR is used to make a permutation in the pixel positions and the XOR operation is done to diffuse the pixel values. Pseudo-random numbers generated by the linear method have been used to perform the bitwise XOR operation.

Chunyan Song and Yulong Qiao [8] proposed a novel image encryption scheme based on DNA encoding and spatiotemporal chaos. The DNA mapping rule is introduced to encode the diffused image that is previously generated from the plain image by primarily diffused with the bitwise XOR operation. The spatiotemporal chaotic system is used to confuse the rows and columns of the DNA encoded image to enhance the encryption.

Prabir and Atal in [9] provided additional protection of the secret data by presenting a scheme that combines substitution as well as transposition techniques. During ciphering, dynamic substitution box (S-BOX) and transposition box (T-BOX) are generated using the secret key and made to vary for each block.

Yu-Guang et al. [10] found that quantum walks (QW) can serve as an excellent key generator thanks to its inherent nonlinear chaotic dynamic behavior. They constructed a novel QW-based image encryption algorithm. The researchers studied the potential application of a famous quantum computation model, i.e., quantum walks (QW) in image encryption.

An image encryption method has been proposed in this paper. This method uses a proportionally large secret key of size (2048) bits, different implementation of the substitution and transposition operations on both image data blocks and secret key block to achieve good diffusion and confusion effects in the image data. Where: diffusion means that changing a single byte in the data blocks of the input image will change many bytes in the data of the produced encrypted image, and confusion means that each binary digit (bit) in the data blocks of the input image should depend on several parts of the key.

All that will contribute to give the users of this algorithm high protection for the image using a large and robust key in a relatively short time for encryption. This will encourage users to use the proposed algorithm because it has the best features when it is compared with other known encryption algorithms.

This paper has been organized into four main sections. Section 1 gives an introduction to the idea of the work and the related recent works in this area. Section 2 presents a detail description of the proposed encryption algorithm with illustrative examples. Section 3 presents some of the experiments carried out to test and evaluate the proposed algorithm and discusses the efficiency of the algorithm through analyzing the results recorded. Section 4 summarizes the conclusions from the work in this paper.

## 2. The proposed image encryption algorithm

Overlapped paths are extracted from the secret key block; these paths are used to implement the substitution and transposition operations on the bytes of the image data blocks. Another way of implementation of the substitution and transposition operations is performed on the secret key block to generate successive different keys. The general steps in the encryption stage of the proposed image encryption algorithm are listed below: To give detailed clarification about how extracting the overlapping paths from the secret key block. A simple example will be given below:

Step 1: Read (from the user) the secret key of length $\geqslant 256$ bytes.
Step 2: Read (from the user) the original image $\boldsymbol{S}$ of length $\boldsymbol{S}_{\text {Length }}$ bytes.
Step 3: Partition $\boldsymbol{S}$ into a set of $\boldsymbol{n}$ blocks of size $(16 \times 16)$, from $\boldsymbol{S} \mathbf{( 0 )}$ to $\boldsymbol{S}(\boldsymbol{n} \mathbf{- 1})$. Where $\boldsymbol{n}=$ $S_{\text {Length }} /(16 \times 16)$
Step 4: Get the first 256 bytes of the user's secret key and represent these bytes as a twodimensional matrix (block) $\boldsymbol{K}$ of size $(16 \times 16)$, i.e., $\mathbf{1 6}$ rows and $\mathbf{1 6}$ columns.
Step 5: For each image data block $S(b)$ from $b=0$ to $n-1$.
Note: $\boldsymbol{i}$ and $\boldsymbol{j}$ indices represent the $\boldsymbol{i}$ th row and $\boldsymbol{j}$ th column in either data block $\boldsymbol{S}(\boldsymbol{b})$ or secret key block $\boldsymbol{K}$.
(i) Extract the overlapping paths of elements from $\boldsymbol{K}$, one path for each $\boldsymbol{K}(i, j)$ element in $K$.
(ii) Perform substitution operation for each $(i, j)$ element in $S(b)$ :

- XORing the $S(b, i, j)$ with all the elements in the path of $\boldsymbol{K}(i, j)$ in $\boldsymbol{K}$.
- Set $S(b, i, j)=255-S(b, i, j)$
- Rotate right the bits of the $\boldsymbol{S}(\boldsymbol{b}, \boldsymbol{i}, \boldsymbol{j})$ number of times equal to the number of bits that have a value $\mathbf{1}$, in each element in the path of elements of $\boldsymbol{K}(i, j)$ in $\boldsymbol{K}$.
(iii) Perform transposition operation for each $(i, j)$ element in $S(b)$ :
- Exchanging sequentially the $S(b, i, j)$ with all the elements $S(b, x, y)$ based on the extracted path of elements of $\boldsymbol{K}(\boldsymbol{i}, \boldsymbol{j})$ in $\boldsymbol{K}$. Where $\boldsymbol{x}$ and $\boldsymbol{y}$ are the row and column for the elements in the path of $\boldsymbol{K}(i, j)$.
(iv) Generate a new secret key $K$ that will use for the next data block $S(b+1)$.
- Perform substitution operation for each $K(i, j)$ by XORing it with the new encrypted value $S(b, i, j)$.
- Perform transposition operation for each $\boldsymbol{K}(i, j)$ by exchanging it with the element $K(x, y)$ as follow:
$\boldsymbol{x}=\boldsymbol{K}(i, j)$ Divide $\mathbf{1 6}$
$y=K(i, j)$ Modulus 16.
Step 6: Construct the encrypted image $\boldsymbol{E}$ of length $\boldsymbol{E}_{\text {Length }}$ from the encrypted blocks $S(\mathbf{0})$ to $S(n-1)$ after completion of the substitution and transposition operations in Step 5.

Assume that the secret key block $\boldsymbol{K}$ be as in Table 1. It should be noted here that the bytes in $\boldsymbol{K}$ are represented as hexadecimal numbers. The left half of each number represents the row number in $\boldsymbol{K}$ and right half of the number represents the column number in $\boldsymbol{K}$.

TABLE 1. Example of $(16 \times 16)$ secret key block $\boldsymbol{K}$.

|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 0 | B9 | D1 | C4 | 8E | 34 | 8F | E7 | 71 | FA | 46 | 4A | 77 | A1 | 78 | FB | 07 |
| 1 | DC | FE | AD | 50 | D1 | D9 | FD | 08 | B3 | 86 | EF | B0 | 8B | 14 | 2 F | 74 |
| 2 | 4 C | FD | A4 | C3 | 07 | 61 | 57 | F5 | 81 | B7 | 1 E | 46 | E8 | CB | 56 | 75 |
| 3 | 25 | 38 | 68 | B7 | 9E | 7 C | 31 | E0 | D3 | BC | DB | AE | 9 F | 37 | E5 | E5 |
| 4 | 16 | D6 | 2B | A4 | D3 | 38 | FF | 7B | A6 | B8 | CF | 19 | BA | B7 | 20 | E7 |
| 5 | 30 | 7A | 8A | 53 | AB | 77 | D5 | CF | CB | C3 | F0 | 4B | 85 | A9 | 55 | 49 |
| 6 | D2 | 38 | 2F | 11 | 9 F | B1 | A1 | ED | 38 | ED | 11 | E6 | AD | 18 | 26 | 1B |
| 7 | 9C | F3 | 7D | 25 | AD | 91 | 39 | 9E | 01 | 5C | D9 | 0F | 48 | B4 | F5 | FD |
| 8 | 63 | 85 | 47 | 81 | 88 | 0E | 47 | 7E | D8 | 9 C | 1A | BE | 66 | EE | 21 | B3 |
| 9 | 9D | D4 | AF | 91 | 2 C | 2 F | 9B | 40 | FF | 91 | 10 | 43 | D7 | 91 | A5 | 84 |
| A | F8 | 13 | 83 | 27 | D0 | A4 | 93 | 8D | AC | FE | 4D | 10 | 77 | 7B | 37 | 12 |
| B | 3A | 26 | 72 | 73 | 05 | 63 | D0 | 33 | E9 | 07 | 82 | F6 | FE | 04 | A7 | 47 |
| C | F7 | 96 | C5 | 52 | 6B | F9 | E2 | F9 | 41 | 21 | E5 | 2 D | E2 | 8B | C8 | 83 |
| D | 5A | FD | 08 | B1 | 1 F | 31 | 7F | 94 | 42 | 4 F | 3D | C8 | 22 | 9C | 8F | A7 |
| E | 57 | 05 | 18 | E0 | B5 | E4 | A6 | 41 | 43 | 79 | 32 | E3 | 7D | A3 | 51 | 5E |
| F | 3 E | FE | 9 C | 00 | FB | F6 | 28 | C5 | 17 | 30 | 10 | 1C | BB | 1A | D6 | A4 |

The extracted path for the element $\boldsymbol{K}(\mathbf{0}, \mathbf{0})$ is the series of elements starting from the index $(\mathbf{0}, \mathbf{0})$ and ends when you reach the element has been visited previously.

$$
(\mathbf{0}, \mathbf{0}) \rightarrow(\mathbf{B}, \mathbf{9}) \rightarrow(\mathbf{0}, \mathbf{7}) \rightarrow(\mathbf{7}, \mathbf{1}) \rightarrow(\mathbf{F}, \mathbf{3})
$$

The elements in the paths of all $\boldsymbol{K}(\boldsymbol{i}, \boldsymbol{j})$ may be overlapped. This will add more diffusion and confusion effects in the encrypted image data blocks. In Table 1 above, the extracted path for the element $\boldsymbol{K}(\mathbf{A}, \mathbf{D})$ contains the following series of elements:

$$
(\mathrm{A}, \mathrm{D}) \rightarrow(7, \mathrm{~B}) \rightarrow(0, \mathrm{~F}) \rightarrow(0,7) \rightarrow(7,1) \rightarrow(\mathrm{F}, 3) \rightarrow(0,0) \rightarrow(\mathrm{B}, 9)
$$

It is clear that the elements $(\mathbf{0}, \mathbf{0}),(\mathbf{B}, \mathbf{9}),(\mathbf{0}, \mathbf{7}),(\mathbf{7}, \mathbf{1}),(\mathbf{F}, \mathbf{3})$ overlap between the two extracted paths of the two elements $\boldsymbol{K}(\mathbf{0}, \mathbf{0})$ and $\boldsymbol{K}(\mathbf{A}, \mathbf{D})$.

To perform the substitution operation on the element (byte) in $\boldsymbol{S}(\boldsymbol{b}, \mathbf{0}, \mathbf{0})$, the following three sub-operations must be implemented (as mentioned in the algorithm):

- Implement set of XORing operations with all elements in the extracted path of $\boldsymbol{K}(\mathbf{0}, \mathbf{0})$ :

$$
\begin{aligned}
& S(b, 0,0)=S(b, 0,0) \text { XOR } K(0,0) \\
& S(b, 0,0)=S(b, 0,0) \operatorname{XOR} K(\mathrm{~B}, 9) \\
& S(b, 0,0)=S(b, 0,0) \operatorname{XOR} K(0,7) \\
& S(b, 0,0)=S(b, 0,0) \text { XOR } K(7,1) \\
& S(b, 0,0)=S(b, 0,0) \text { XOR } K(\mathrm{~F}, 3)
\end{aligned}
$$

- Set $S(b, 0,0)=255-S(b, 0,0)$.
- Implement set of Rotate Right operations on the bits of the $\boldsymbol{S}(\boldsymbol{b}, \mathbf{0}, \mathbf{0})$ as follow:

Rotate Right $[\boldsymbol{S}(\boldsymbol{b}, \mathbf{0}, \mathbf{0})$, Number of $\operatorname{Ones}(\boldsymbol{K}(\mathbf{0}, \mathbf{0}))$ ]
Rotate Right $[\boldsymbol{S}(\boldsymbol{b}, \mathbf{0}, \mathbf{0})$, Number of $\operatorname{Ones}(\boldsymbol{K}(\mathbf{B}, \mathbf{9}))]$
Rotate Right[ $\boldsymbol{S}(\boldsymbol{b}, \mathbf{0}, \mathbf{0})$, Number of $\operatorname{Ones}(\boldsymbol{K}(\mathbf{0}, \mathbf{7}))$ ]
Rotate Right $[\boldsymbol{S}(\boldsymbol{b}, \mathbf{0}, \mathbf{0})$, Number of $\operatorname{Ones}(\boldsymbol{K}(\mathbf{7}, \mathbf{1}))]$
Rotate Right[ $\boldsymbol{S}(\boldsymbol{b}, \mathbf{0}, \mathbf{0})$, Number of $\operatorname{Ones}(\boldsymbol{K}(\mathbf{F}, \mathbf{3}))]$.


Figure 1. (a) Original image. (b) Diffusion effect. (c) Confusion effect.

To perform the transposition operation on the element (byte) in $\boldsymbol{S}(\boldsymbol{b}, \mathbf{0}, \mathbf{0})$, sequentially exchange $\boldsymbol{S}(\boldsymbol{b}, \mathbf{0}$, $\mathbf{0})$ with all elements in the extracted path of $\boldsymbol{K}(\mathbf{0}, \mathbf{0})$.

$$
\begin{aligned}
S(b, 0,0) & \leftrightarrow S(b, \mathrm{~B}, 9) \\
S(b, \mathrm{~B}, 9) & \leftrightarrow S(b, 0,7) \\
S(b, 0,7) & \leftrightarrow S(b, 7,1) \\
S(b, 7,1) & \leftrightarrow S(b, \mathrm{~F}, 3) .
\end{aligned}
$$

To re-generate the original image $\boldsymbol{S}$, in the decryption stage, the same steps mentioned above are applied in inverse sequence on the encrypted image $\boldsymbol{E}$.

## 3. The results of experiments and security analysis

The measurements of key space, key sensitivity, and statistical analysis have been used to conduct performance comparison between the proposed algorithm and some well-known encryption algorithms like Data Encryption Standard (DES) and Advanced Encryption Standard (AES). The recorded results have been analyzed to check the quality level of protection achieved in the proposed encryption algorithm. The implementation of the proposed encryption algorithm is tested using images of different size.

### 3.1. Diffusion and confusion effects

The implementation of the substitution and transposition operations that are performed in the proposed encryption algorithm causes good diffusion and confusion effects in the original image data. Figure 1 depicts the effects of the substitution and transposition operations separately.

### 3.2. Key space analysis

It is good for any encryption algorithm that uses a large key as much as possible. The number of bits in the key (key space) should be large enough to maximize the difficulty in using the brute-force attack to break the encrypted data and to produce an efficient encryption system. The secret key size that is used in the proposed encryption system has been represented as a two-dimensional matrix $(\mathbf{1 6} \times \mathbf{1 6})$ of bytes. This means that the key size equals $(\mathbf{1 6} \times \mathbf{1 6} \times \mathbf{8}$ bits $)=\mathbf{2 0 4 8}$, this size is larger than the key size used in the widely known encryption algorithms (DES and AES). This makes the guessing operation for the key by the attacker is hard. In addition to that, the proposed encryption system generates and uses a number of successive different keys to encrypt each data block of the image.


Figure 2. (a) Original image. (b) Decrypted image using wrong key (One bit changed).


Figure 3. (a) Original image and its histogram. (b) Encrypted image and its histogram.

### 3.3. Key sensitivity

The sensitivity feature of the key that is used in the encryption system means that any changes in the key, even in one bit, will lead to fail to decrypt the encrypted image to re-generate the original image. The proposed encryption system has been tested by changing one bit in the key and then using it to decrypt the encrypted image. The newly generated image, through the decryption operation, using the wrong key is completely different from the original image as shown in Figure 2. This means that the proposed encryption system is highly sensitive to any small changes in the key.

### 3.4. Statistical analysis

The statistical attack can be done by analyzing the histogram of both original and encrypted image. The strength of the encryption system can be proven by producing encrypted image has different histogram than the histogram of the original image. Figure 3 shows the histogram for both the original and the encrypted image

TABLE 2. Original images and their encrypted images using AES, DES, and the proposed algorithm.

respectively, it is clear they are largely different. If the attackers use the statistical analysis of the histograms, they are unable to get any useful information from these histograms that help them to break the encrypted image.

To stand on the promising results of the proposed image encryption system, three encryption systems (the proposed encryption system, data encryption standard (DES) and advanced encryption standard (AES) systems) have been implemented (on same computer machine) using different images, the results of these tests have been recorded in Tables 2 and 3. Signal to Noise Ratio ( $S N R$ ), Peak Signal to Noise Ratio ( $P S N R$ ), Normalized Mean Absolute Error (NMAE) and Time of Encryption (Time) have been used to check the performance and the quality level of protection that can achieve in each of the encryption systems used in these tests.

Analysis of the results recorded in the experiments helped to stand on the strengths and weaknesses of the proposed algorithm. The strengths of the proposed algorithm been achieved in:

- High degree of confusion and diffusion effects in the encrypted image data appeared which makes difficult for attackers to identify the original image using the Human Visual System (HVS).
- Use large key (2048 bits) in the extraction of a map of the overlapped paths helped to make the proposed algorithm has high sensitivity to any small changes in the key.
- Success in making very significant changes in the histogram of colors in the encrypted image compared with the histogram of the original image.
- The time required to complete the encryption stage is significantly less than other reputable methods.

But here we can say that the main weakness in the proposed algorithm is in the selection the initial key by the user. Because the bytes values in the key must be random as much as possible to get long and high-overlapped

Table 3. Results of tests.

| Image (Width $\times$ Height) | Measurement | Encryption method |  |  |
| :---: | :---: | :---: | :---: | :---: |
|  |  | AES | DES | Proposed method |
| Palm trees $(245 \times 206)$ | $S N R_{d b}$ | 3.98 | 3.98 | 3.94 |
|  | $P S N R_{d b}$ | 8.50 | 8.52 | 8.69 |
|  | NMAE $_{\%}$ | 63.21 | 63.08 | 61.25 |
|  | Time $_{\text {sec }}$ | 25.77 | 25.98 | 1.89 |
| Nemo-Dory $(256 \times 192)$ | $S N R_{d b}$ | 3.05 | 3.03 | 2.97 |
|  | $P S N R_{d b}$ | 7.22 | 7.21 | 7.75 |
|  | $N M A E_{\%}$ | 71.16 | 71.10 | 67.16 |
|  | Time $_{\text {sec }}$ | 28.14 | 24.33 | 1.90 |
| Baboon $(128 \times 128)$ | $S N R_{d b}$ | 4.16 | 4.19 | 4.13 |
|  | $P S N R_{d b}$ | 7.60 | 7.62 | 8.90 |
|  | $N M A E_{\%}$ | 64.62 | 64.46 | 63.66 |
|  | Time $_{\text {sec }}$ | 2.74 | 2.68 | 0.62 |

paths of the key. But the algorithm tried to limit the impact of the weak randomization through the generation of new keys periodically.

## 4. Conclusion

Using a large key size $=2048$ bits, generate successively different keys and performing substitution and transposition operations on the image data based on the extracted map of overlapped paths from the secret key; all these properties in the proposed image encryption method produced a good degree of confusion and diffusion in the encrypted image. The comparison of the performance and the security analysis between the proposed encryption system and the well-known encryption methods (DES and AES) showed that:

- The proposed encryption method succeeded in achieving very close results of security as in DES and AES methods based on the visual and statistical tests, SNR, PSNR and NMAE.
- The proposed encryption method is superior to other methods in decreasing the time required for encryption.
- The proposed encryption method uses the key size larger than the key size used in other well-known encryption methods (DES and AES).

Therefore, the proposed encryption method can be used effectively in the field of information security to achieve high protection for images.

## References

[1] G.A. Sathish Kumar, K. Bhoopathy Bagan and V. Vivekanand, A Novel Algorithm for Image Encryption by Integrated Pixel Scrambling Plus Diffusion [IISPD] Utilizing Duo Chaos Mapping Applicability in Wireless Systems. Procedia Comput. Sci. 3 (2011) 378-387.
[2] M.W. Salim and Nasharuddin Zainal, Rapid Encryption Method Based on AES Algorithm for Grey Scale HD Image Encryption. Procedia Tech. 11 (2013) 51-56.
[3] M. Chaumonta, W. Puech and C. Lahanier, Securing Color Information of an Image by Concealing the Color Palette. J. Syst. Softw. 86 (2013) 809-825.
[4] S. Megha, Reader and S. Kaminee, Image Encryption and Decryption Using Selective Block Encryption Technique. Int. J. Comput. Sci. Eng. Tech. (IJCSET) 5 (2014) 941-946.
[5] M. Al-Husainy, A Novel Encryption Method for Image Security. Int. J. Secur. Appl. 6 (2012) 1-8.
[6] K. Kuppusamy and K. Thamodaran, Optimized Hybrid Security Mechanism for Image Authentication and Security using PSO. Int. J. Netw. Secur. Appl. (IJNSA) 5 (2013) 115-128.
[7] T. Sivakumar and R. Venkatesan, A Novel Image Encryption Approach using Matrix Reordering. WSEAS Trans. Comput. 12 (2013) 407-417.
[8] S. Chunyan and Q. Yulong, A Novel Image Encryption Algorithm Based on DNA Encoding and Spatiotemporal Chaos. Entropy 17 (2015) 6954-6968.
[9] Kr.N. Prabir and C. Atal, A Secure Symmetric Image Encryption Based on Bit-wise Operation. Int. J. Image Graph. Signal Process. 6 (2014) 30-38.
[10] Y. Yu-Guang, P. Qing-Xiang, S. Si-Jia and X. Peng, Novel Image Encryption based on Quantum Walks. Scientific Rep. 5 (2015) 7784.

Communicated by S. Mesnager.
Received March 28, 2016. Accepted October 6, 2016.


[^0]:    Keywords and phrases. Hexadecimal, XOR, transposition, substitution, information security.
    ${ }^{1}$ Department of Computer Science, Faculty of Information Technology, Middle East University, Amman, Jordan. dralhusainy@gmail.com; mal-husainy@meu.edu.jo; dralhusainy@yahoo.com

