Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.18576/amis/160619

Building a Secure Image Cryptography System using Parallel Processing and Complicated Dynamic Length Private Key

Mua'ad Abu-Faraj[1,*], *Abeer Al-Hyari*[2], *Bilal Al-Ahmad*[1], *Ziad Alqadi*[3], *Basel Ali* [4] *and Abdullah Alhaj* [5]

[1]Department of Computer Information Systems, The University of Jordan, Aqaba 77110, Jordan
[2]Electrical Engineering Department, Al-Balqa Applied University, As Salt 19117, Jordan
[3]Computers and Networks Engineering Department, Al-Balqa Applied University, Amman 15008, Jordan
[4]Accounting and Finance Department, Applied Science University, Al Eker 623, Kingdom of Bahrain
[5]Department of Information Technology, The University of Jordan, Aqaba 77110, Jordan

Received: 12 Jul. 2022, Revised: 2 Sep. 2022, Accepted: 6 Sep. 2022
Published online: 1 Nov. 2022

Abstract: A method of color images cryptography will be introduced, programmed, and tested. The proposed method is based on using a digital color image as an image key; this image is to be kept secret without transmission. The proposed method will provide a high level of images protection based on the complicated and complex private key used in cryptography, this key will be changed when replacing the image key, or changing the data block size, or changing the color channel. The proposed method will be compared with other standard methods of data cryptography, and it will be shown how this method will improve the efficiency of data cryptography by minimizing the encryption-decryption time, the obtained results will be compared with the standard method of data cryptography to show the speedup achieved by the proposed method. It will be shown how to execute the proposed method in parallel, 2, 4, and 8 threads will be used to execute the method and the associated speedup will be calculated. The proposed method will protect the data by providing a high level of security, this can be achieved by using a variable-length private key, the private key length and content will depend on the selected image key, selected color matrix, and the selected block size. The block size used in the proposed method will be variable and it will be shown that the proposed method will satisfy the quality requirements by providing good value for Mean Square Error (MSE), and Peak Signal to Noise Ratio (PSNR)

Keywords: Image_key; PK; block size; MSE; PSNR

1 Introduction

Color images are considered one of the most widespread digital data types used in many critical vital applications. This spread is due to several reasons, the most important of which are [1,2,3,4]:

- Ease of obtaining a digital color image at no cost.
- The high image size can be used for different purposes and take advantage of the digital data contained in the digital image.
- The text in the entries may be of any length.
- The ease of processing a color digital image is usually represented by a three-dimensional matrix, one for each of the three colors (red, green, and blue) and as shown in Figure 1.

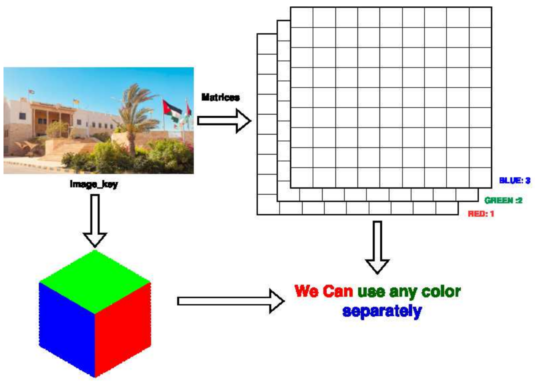


Fig. 1: Color image representation

[*] Corresponding author e-mail: m.abufaraj@ju.edu.jo

- The possibility of dealing with each color's matrix and parts of the image separately.
- The possibility of adjusting the size of the image to suit any other size, whether by reducing the size or increasing it, is shown in Figure 2.

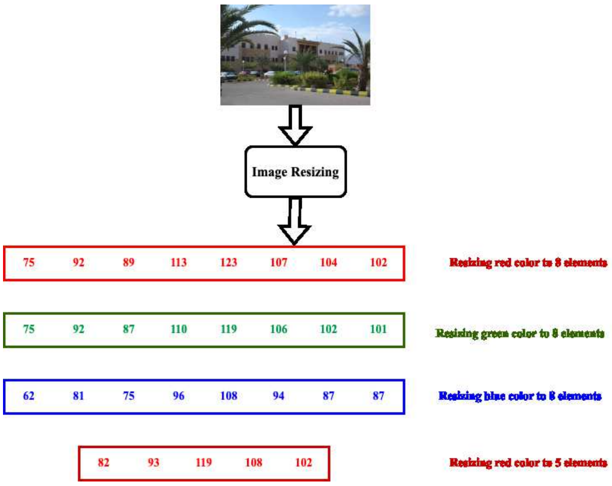


Image Resizing
75 92 89 113 123 107 104 102
Resizing red color to 8 elements
75 92 87 110 119 106 102 101
Resizing green color to 8 elements
62 81 75 96 108 94 87 87
Resizing blue color to 8 elements
82 93 119 108 102
Resizing red color to 5 elements

Fig. 2: Image resizing

The possibility of using digital images to protect all types of digital data using hiding (data steganography), encryption, and decryption processes (data cryptography). Color digital images require protection from the danger of tampering, intruders, or data thieves for several reasons [5,6,7,8], the most important of these reasons. First, the digital image can be of a private or confidential nature. Second is the possibility of the digital image carrying personal data. Third, frequent use of digital images in applications requires more protection. Fourth, the circulation of digital images might provide the possibility of accessing them by unrelated or unauthorized persons or entities. Data cryptography is one method used to protect digital color images; it can be performed as shown in Figure 3 by using the private key (PK) and manipulating several arithmetical and logical operations using the data to be encrypted and PK.

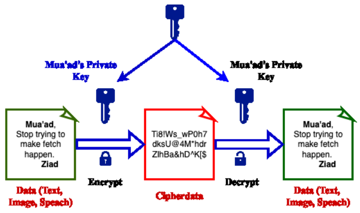


Fig. 3: Data cryptography process

2 Related Work

Multiple methods [9,10,11,12,13,14] encrypt and decrypt data, including digital images. When choosing a specific method to protect data, this method must achieve the following conditions:

- The method should be secure to provide a high degree of data protection, achieved using a private secret key that is difficult to hack, know or guess.
- The method works to destroy and distort the data when encrypting so that the data becomes useless or difficult to understand for any unauthorized third party and works to return the original data without change when decrypting. The percentage of destruction can be measured using the quality parameter MSE or PSNR; the MSE value between the encrypted data and the original one must be very high (PSNR must be very low, while MSE between the original data and the decrypted one must equal zero (PSNR must be infinite) (see equations 1, and 2) [9,10].

$$MSE = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}(x_{ij}-y_{ij})^2 \tag{1}$$

where m is the number of rows in cover image, n is the number of columns in cover image, x_{ij} is the pixel value from cover image, and y_{ij} is the pixel value from stego image [10,11,12].

$$PSNR = 10\log_{10}\frac{[MAX_I]^2}{MSE_t} \tag{2}$$

where MAX_I is the maximum signal value that exists in our original "known to be good" image [9].

- MSE and PSNR are good parameters to measures the changes between the source image and the encrypted/decrypted image, the changes are due to applying encryption-decryption process, If MSE equal zero, then PSNR is equal infinite, this means that there is no changes in the decrypted image, and the decrypted image is identical to the source image [10].
- The method should be flexible, so it is easy to change the private key or modify it by increasing or decreasing its length. It is also easy to alter the data block length used in encryption and decryption[11].
- The method should be highly efficient by reducing the encryption time and decryption times to the lowest possible value, thus increasing the method throughput (number of bytes treated in a unit of time) [11].
- The method should be easy and feasible, programmatically or hardware [12].

Multiple methods are now used to protect data based on international standards, including Data Encryption standard (DES), Triple DES (3DES), Advanced Encryption Standard (AES), and blowfish (BF) [15,16,17,18,19]. These methods share many inefficient and unsecured features; Table 1 shows these methods' main features.

Faster methods were introduced; these methods were used to minimize the encryption-decryption times and to maximize the throughput of data cryptography; in [30], the authors provided a robust and fast image encryption scheme based on a mixing technique. In [31], the authors provided a cosine-transform-based chaotic system for image encryption. In contrast, in [32], the authors introduced a novel image encryption algorithm based on a polynomial combination of chaotic maps and dynamic function generation. In [33], the authors introduced Multiple-image Encryption Algorithm Based on DNA Encoding and Chaotic Systems, while in [34], the authors produced multiple-image encryption with bit-plane decomposition and chaotic maps; these methods provided good quality.

In [21], a comparative analysis of DES, 3DES, and AES was done, and the performance and throughputs of these methods were calculated; the throughput will rapidly decrease when the data size increases; thus, using these methods for image encryption-decryption will be not efficient. In [18,22,23,29,35], a performance analysis or blowfish method of data cryptography was done; the results showed that using this method will increase the efficiency compared with DES and AES methods, but using it for image encryption-decryption requires enhancements to increase the cryptography process throughput.

3 The Proposed Method

The proposed method shown in Figure 4 provides a high degree of data security and protection through the use of a color image to generate the private keys used in the encryption and decryption process that are difficult to penetrate or guess for the following reasons:

–The key image is determined by agreement between the sender and receiver and is kept secretly and without resorting to sending or circulating it.
–The ability to change the key image at any time and if needed.
–Changing the data block size and the selected color matrix for resizing will change the length and contents of the private key as shown in Figure 5 to Figure 7.
–Changing the image key will change the contents of the private key (see Figure 5 and Figure 6).

The Feistel functions use a rotate operation; the selected number of rotating digits can change from 0 to 7 in the encryption phase and from 7 to 0 in the decryption phase. The proposed method of color image encryption can be implemented, as shown in Figure 8, by applying the following steps:

–Step 1: The initialization step includes the following substeps: select the image key, select the color matrix to be used to generate PK, select the data block size in bytes, and resize the color matrix to meet the block size; the resized image will be used as a PK, and reshape the image to be encrypted into a one-row matrix.
–Step 2: For each block of data, apply the Feistel function for each byte, XOR the results with the associated byte from the PK, then use the second Feistel function.
–Step 3: Reshape the encrypted data to the 3D matrix to get the encrypted image. The decryption phase can be implemented reversely, as shown in Figure 8.

4 Implementation and Experimental Results

The proposed method was programmed using Matlab code; the program was executed several times using an I7 multicore processor with 8 G byte RAM. Figure 9 shows an output example of the proposed method execution.

Images shown in Table 2 were taken, encrypted decrypted using various initial stages; Table 3 shows the obtained results using image 4 as an image_key, block size=16 bytes:

The same images were taken, encrypted-decrypted using image 2 (small image) as an image_key, with block size=100 bytes; the results are shown in Table 3.

The selected images were encrypted-decrypted, varying the block size. Table 4, Table 5, and Table 6 show the experimental results.

Based on the exciting findings, the encryption time increases as the number of keys increases. The results have shown how the proposed method would improve the efficiency of data cryptography by reducing the encryption-decryption time. The obtained results are compared with the standard method of data cryptography to show the speedup achieved by the proposed method. The traditional cryptography methods that were examined are (DES, 3DES, AES, and BF). The proposed method satisfies the requirements for image quality in the encryption and decryption phases [14,?,15].

For comparison purposes, the standard methods of data encryption-decryption were implemented using the same selected images; Table 7 shows the obtained experimental results compared to the methods (DES and 3DES). Also, Table XI compares AES and BF methods and the proposed approach (MPK).

The proposed method was implemented using a multithreading environment by executing the code of the proposed method using Matlab pool as shown in Figure 10, varying the number of threads; the obtained results are shown in Table 8 compared to the studies [24,25,26].

The average speedup by implementing 8 threads outperforms the other executed threads. Using one thread is more costly and more time-consuming. Also, increasing the number of threads increases the speed, as shown in Table 8. The multithread approach [27] affects the encryption systems.

Table 1: Cryptography methods main features [1,3,4,5,6,7,36]

Algorithm parameter	Data Encryption standard (DES)	Triple DES (3DES)	Advanced Encryption Standard (AES)	Blowfish
PK length (bit)	56 (fixed)	112, 168 (fixed)	128, 192, 256 (fixed)	32-448 (fixed)
Block size (bit)	64 (fixed)	64 (fixed)	128 (fixed)	64 (fixed)
Ability to deal with images	Difficult	Difficult	Difficult	Difficult
Encryption quality	Excellent: High MSE and low PSNR	Excellent: High MSE and low PSNR	Excellent: High MSE and low PSNR	Excellent: High MSE and low PSNR
Decryption quality	Excellent: Zero MSE and infinite PSNR	Excellent: Zero MSE and infinite PSNR	Excellent: Zero MSE and infinite PSNR	Excellent: Zero MSE and infinite PSNR
Efficiency	Slow	Slow	Slow	Moderate
Attack	Brute force attack	Brute force attack, Known plaintext, Chosen plaintext	Side channel attack	Dictionary attack
Structure	Feistily	Feistel	Substitution-Permutation	Feistel
Block cipher	Binary	Binary	Binary	Binary
Rounds	16 (fixed)	48 (fixed)	10,12,14 (fixed)	16 (fixed)
Flexibility to modification	no	yes	yes	yes
Simplicity	no	no	no	no
Security level	Adequate	Adequate	Excellent	Excellent
Throughput	Low	low	Low	Moderate

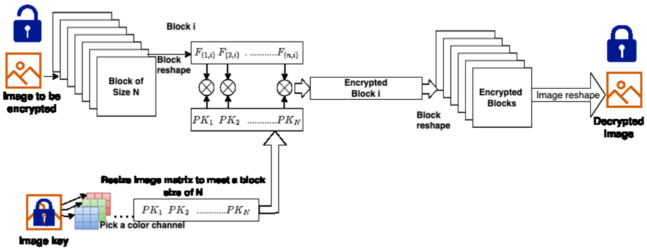


Fig. 4: The proposed method

Table 2: Results using image 4 as an image_key, block size=16 bytes

Image number	Dimension	Size (byte)	MSE	PSNR	Encryption time (second)
1	151x333x3	150849	1.4272e+004	15.1648	044650
2	152x171x3	77976	1.5070e+004	14.6209	0.037891
3	360x480x3	518400	1.3756e+004	15.5326	0.094956
4	1071x1600x3	5140800	1.2969e+004	16.1223	0.767682
5	981x1470x3	4326210	1.2915e+004	16.1640	0.614167
6	165x247x3	122265	1.2539e+004	16.4590	0.043193
7	360x480x3	518400	1.5040e+004	14.6404	0.098163
8	183x275x3	150975	1.3846e+004	15.4681	0.044476
9	183x275x3	150975	1.2767e+004	16.2791	0.042990
10	201x251x3	151353	1.4221e+004	15.2005	0.042769
11	600x1050x3	1890000	1.3557e+004	15.6790	0.299206
12	1144x1783x3	6119256	1.2623e+004	16.3927	0.879196

5 Discussion

From the obtained experimental results, we can raise many significant findings. First, the proposed method is highly secure; the private key is variable, and the length and the contents of PK depend on the selected image_key, the selected color matrix for resizing, and the block size. Second, the data block size is variable. Third, the encryption time will decrease when selecting image_key with a small size and using data blocks with bigger sizes, as shown in Figure 11. Fourth, increasing block size will decrease encryption time as in the study [28]; the optimal block size for encrypting and decrypting [29] the selected images was 800 bytes. Fifth, the proposed method provides a significant speedup of the process of data cryptography compared with standard methods. In

Table 3: Results using image 2 as an image_key, block size=100 bytes

Image Number	Dimension	Size (byte)	MSE	PSNR	Encryption time (second)
1	151x333x3	150849	2.7697e+004	8.5344	0.009153
2	152x171x3	77976	3.5373e+004	6.0882	0.004588
3	360x480x3	518400	2.1046e+004	11.2808	0.016747
4	1071x1600x3	5140800	1.6403e+004	13.7731	0.137417
5	981x1470x3	326210	1.6161e+004	13.9219	0.118353
6	165x247x3	122265	1.1708e+004	17.1454	0.007986
7	360x480x3	518400	2.3129e+004	10.3367	0.018330
8	183x275x3	150975	2.1813e+004	10.9226	0.006040
9	183x275x3	150975	1.6117e+004	13.9491	0.006559
10	201x251x3	151353	2.4935e+004	9.5848	0.007003
11	600x1050x3	18000	2.3488e+004	10.1830	0.060182
12	1144x1783x3	6119256	6.9825e+003	22.3136	0.166045

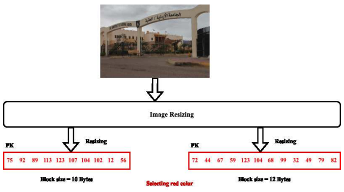


Fig. 5: Selecting the red color of image_key to generate PK

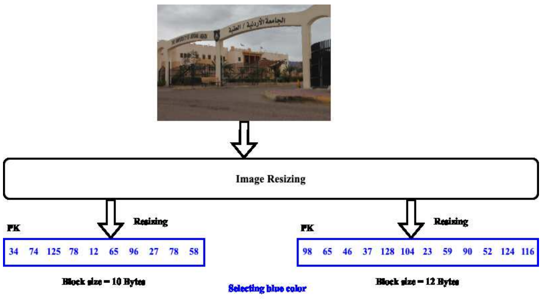


Fig. 6: Selecting the blue color of image_key to generate PK

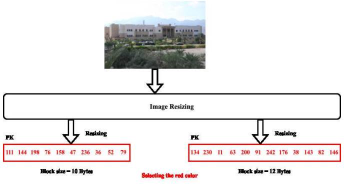


Fig. 7: Selecting the red color of image_key to generate PK

Table 4: Encrypting-decrypting image 2, image 4 is an image_key

BLS (byte) (number of keys)	MSE	PSNR	Encryption time (second)
16	1.5070e+004	14.6209	0.037891
32	1.5373e+004	14.4216	0.030081
40	1.5809e+004	14.1420	0.027030
100	1.5393e+004	14.4083	0.024358
500	1.5369e+004	14.4241	0.025259
800	**1.5453e+004**	**14.3699**	**0.022982**
1000	1.5265e+004	14.4923	0.024595

Table 5: Encrypting-decrypting image 12, image 4 is an image_key

BLS (byte) (number of keys)	MSE	PSNR	Encryption time (second)
16	1.2623e+004	16.3927	0.857091
32	1.1618e+004	17.2218	0.502785
40	1.2043e+004	16.8631	0.382784
100	1.1763e+004	17.0983	0.175020
500	1.1727e+004	17.1285	0.074264
800	**1.1741e+004**	**17.1172**	**0.064809**
1000	1.1688e+004	17.1622	0.065107

Table 6: Encrypting-decrypting image 3, image 4 is an image_key

BLS (byte) (number of keys)	MSE	PSNR	Encryption time (second)
16	1.3756e+004	15.5326	0.094956
32	1.3186e+004	15.9563	0.059891
40	1.3521e+004	15.7051	0.058479
100	1.3362e+004	15.8237	0.035774
500	1.3358e+004	15.8265	0.028064
800	**1.3358e+004**	**15.8268**	**0.026850**
1000	1.3315e+004	15.8589	0.029128

addition, we can observe that the proposed method can be easily implemented in parallel using a multithreading system with various threads. There is a significant speedup using two or more threads. Also, the proposed method satisfies the image quality requirement by providing excellent value for MSE and PSNR in both the encryption and decryption phases. The proposed method

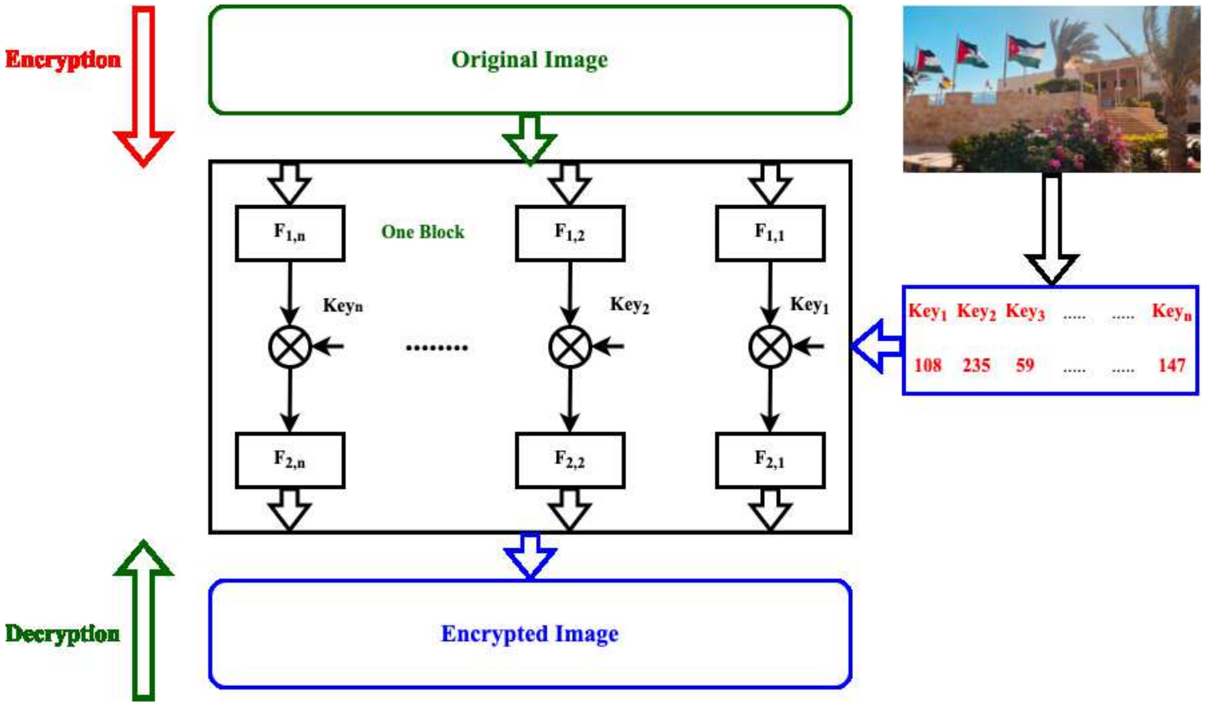


Fig. 8: Selecting the blue color of another image_key to generate PK

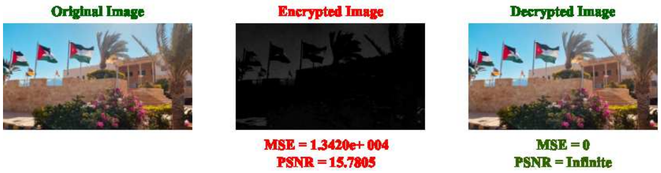
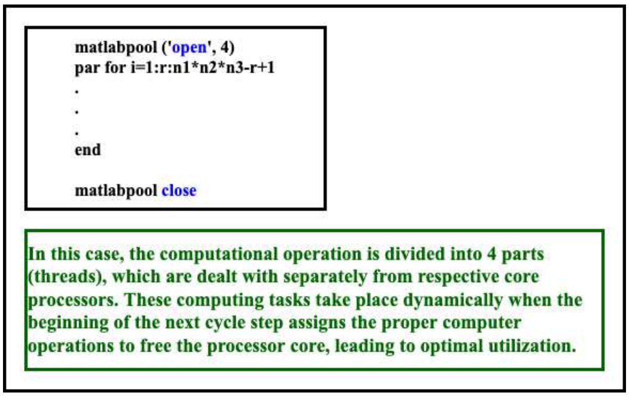


Fig. 9: Selecting the blue color of another image_key to generate PK



```
matlabpool ('open', 4)
par for i=1:r:n1*n2*n3-r+1
.
.
.
end

matlabpool close
```

In this case, the computational operation is divided into 4 parts (threads), which are dealt with separately from respective core processors. These computing tasks take place dynamically when the beginning of the next cycle step assigns the proper computer operations to free the processor core, leading to optimal utilization.

Fig. 10: Multithreading Matlab pool

can be used for various data types, including color images and text files.

The proposed method results were compared with the methods proposed in [30,31,32,33,34], and the proposed method show a significant speedup as show in Table 9:

The proposed method adds the following improvements to the standard techniques of data cryptography:

1. It maintains a higher security level. The private key length is variable
2. The number of generated keys will equal the block size; the private key contents depend on the selected image_key and the block size.
3. One round is required, and expanding the number of rounds is straightforward.
4. Using the proposed approach would speed up the cryptography process.

Table 7: Methods comparisons

Image size(byte)	Encryption time(second)				
	DES	3DES	AES	BF	MPK (BLS=16 byte=128 bits, rounds=10)
150849	0.1093	0.1249	0.1008	0.0603	0.009153
77976	0.0572	0.0647	0.0522	0.0313	0.004588
518400	0.3761	0.4292	0.3456	0.2067	0.016747
5140800	3.6998	4.2546	3.4248	2.0452	0.137417
4326210	3.1138	3.5808	2.8822	1.7213	0.118353
122265	0.0884	0.1015	0.0818	0.0490	0.007986
518400	0.3739	0.4294	0.3457	0.2064	0.018330
150975	0.1085	0.1253	0.1012	0.0607	0.006040
150975	0.1089	0.1252	0.1012	0.0607	0.006559
151353	0.1092	0.1257	0.1013	0.0604	0.007003
1890000	1.3607	1.5646	1.2595	0.7521	0.060182
6119256	4.4038	5.0648	4.0767	2.4345	0.166045
Average	1.1591	1.3326	1.0728	0.6407	0.0465
Throughput (K byte)	1356.3	1179.7	1465.4	2453.7	33808
Speedup of the Proposed method	**24.9266**	**28.6581**	**23.0708**	**13.7784**	**1.0000**

Table 8: Proposed method implementation on multithreading environment

Image number	1 thread time	2 threads time	4 threads time	8 threads time
1	0.044650	0.0263	0.0123	0.0068
2	0.037891	0.0234	0.0105	0.0058
3	0.094956	0.0519	0.0259	0.0142
4	0.767682	0.3998	0.2069	0.1142
5	0.614167	0.3232	0.1660	0.0915
6	0.043193	0.0259	0.0119	0.0066
7	0.098163	0.0536	0.0267	0.0147
8	0.044476	0.0254	0.0122	0.0067
9	0.042990	0.0246	0.0118	0.0064
10	0.042769	0.0244	0.0117	0.0064
11	0.299206	0.1662	0.0818	0.0447
12	0.879196	0.4440	0.2345	0.1303
Average	0.2508	0.1324	0.0677	0.0374
Speedup	**1.0000**	**1.8943**	**3.7046**	**6.7059**

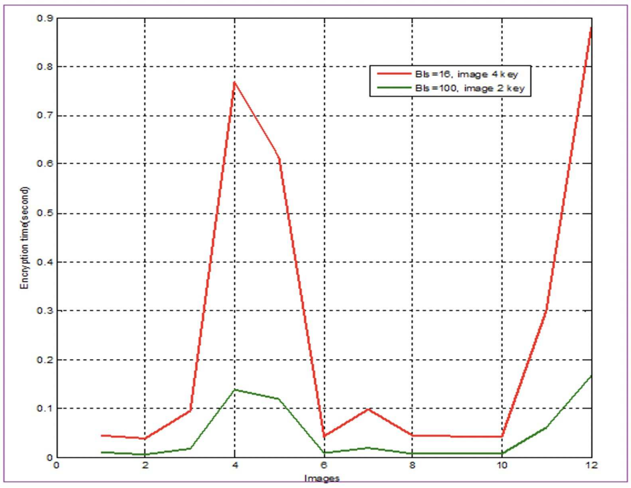


Fig. 11: The relationship between Encryption time and image size

6 Conclusions

A method of color image cryptography was introduced, programmed, and executed. This method can encrypt-decrypt any data, including color images. The proposed method provides a high-security level; this can be achieved depending on the selected image_key, the color matrix, and the data block size, which form the length and contents of the secret private key. It was shown that the private key is not fixed and changed dynamically when the image key or the selected color and data block

Table 9: Throughput comparisons

Method	Throughput (K bytes per second)	Speed up of the proposed method
Ref. [21]	888.8867	40.4877
Ref. [22]	638.4082	56.3730
Ref. [23]	911.0352	39.5034
Ref. [24]	361.4102	99.5794
Ref. [25]	384.9609	93.4874
Proposed*	35989	1.0000
For image 12 throughput= 6119256/(0.166045*1024)= 35989 K bytes per second		

size changed. The proposed method enhances the data cryptography process, providing a good speedup compared with other standard methods; this speedup can be increased by implementing the proposed method using a multithreading environment. The obtained MSE and PSNR values were acceptable for both the encryption and decryption phases. The proposed method can be in the future implemented using hardware; here, a particular processor can be efficiently designed to handle the operations of the proposed method.

References

[1] M. Abu-Faraj, Z. Alqadi, B. Al-Ahmad K. Khaled, B. Ali, A Novel Approach to Extract Color Image Features using Image Thinning, *Applied Mathematics & Information Sciences*, **16**, 4, 65-94 (2022). https://doi.org/10.3390/sym14040664.

[2] M. Abu-Faraj, Z. Alqadi, M. Zubi, Creating Color Image Features Based on Morphology Image Processing, *Traitement du Signal*, **39**, 3, 105-112 (2022). https://doi.org/10.18280/ts.390304.

[3] M. Abu-Faraj, M. Zubi, Analysis and implementation of kidney stones detection by applying segmentation techniques on computerized tomography scans, *Italian Journal and Applied Mathematics*, **43**, 451-458 (2020).

[4] M. Abu-Faraj, A. Al-Hyari, Z. Alqadi, A Complex Matrix Private Key to Enhance the Security Level of Image Cryptography, *Symmetry*, **14**, 4, 664-678 (2022). https://doi.org/10.3390/sym14040664.

[5] A. K. Mandal, C. Parakash, A. Tiwari, Performance evaluation of cryptographic algorithms: DES and AES, *2012 IEEE Students' Conference on Electrical, Electronics and Computer Science*, **2012**, 1-5 (2012).

[6] S. M. Dudhani, S.S. Lomte, Performance Analysis of Data Encryption Algorithms for Secure EHR Transmission, *International Journal of Computer Sciences and Engineering*, **7**, 2, 363-366 (2019). https://doi.org/10.26438/ijcse/v7i2.363366

[7] S. Sharma, L. Kumar, H. Sharma, Encryption of an audio file on lower frequency band for secure communication, *International Journal of Advanced Research in Computer Science and Software Engineering*, **3**, 7, 79-84 (2013).

[8] S. Sharma, L. Kumar, H. Sharma, Steganography A Sin qua non for Diguised Communication, *International Journal of Innovative Research in Advanced Engineering (IJIRAE)*, **1**, 184-191 (2014).

[9] S.P. James, S. N.George, P.P. Deepthi, *Secure selective encryption of compressed audio*, in Proc. 2013 Annual International Conference on Emerging Research Areas and 2013 International Conference on Microelectronics, Communications and Renewable Energy, 1-6, (2013).

[10] S. Pavithra, E. Ramadevi, Throughput analysis of symmetric algorithms, *International Journal of Advanced Networking and Applications*, **4**, 2, 1574-1581 (2012).

[11] S. Sharma, L. Kumar, H. Sharma, Evaluating The Performance of Symmetric Encryption Algorithms, *International Journal of Network Security*, **10**, 3, 216-222 (2010).

[12] M. Abu-Faraj, K. Aldebei, Z. Alqadi, Simple, Efficient, Highly Secure, and Multiple Purposed Method on Data Cryptography, *Traitement du Signal*, **39**, 1, 173-178 (2022). https://doi.org/10.18280/ts.390117.

[13] M. Abu-Faraj, Z. Alqadi, Simple, Efficient, Using Highly Secure Data Encryption Method for Text File Cryptography, *International Journal of Computer Science & Network Security*, **21**, 12, 53-60 (2021).

[14] M. Abu-Faraj, Z. Alqadi, Rounds Reduction and Blocks Controlling to Enhance the Performance of Standard Method of Data Cryptography, *International Journal of Computer Science & Network Security*, **21**, 12, 448-465 (2021).

[15] S. Sharma, P. Pateriya, A study on different approaches of selective encryption technique, *International Journal of Computer Science & Communication Networks*, **2**, 6, 658-668 (2012).

[16] D. AbdulMinaam, H. Abdual-Kader, H. Hadhoud, Evaluating the Effects of Symmetric Cryptography Algorithms on Power Consumption for Different Data Types, *International Journal of Network Security*, **11**, 2, 78-87 (2010).

[17] R. Kumar, B. Saini, S. Kumar, N. Kurukshetra, A Novel Approach to Blowfish Encryption Algorithm, *International Journal of Advanced Research in Science, Engineering and Technology (IJARSET)* , **1**, 62-67 (2014).

[18] S. Rizvi, S. Hussain, N. Wadhwa, *Performance analysis of AES and TwoFish encryption schemes*, in Proc. 2011 International Conference on Communication Systems and Network Technologies, 76-79, (2011).

[19] P. Singh, K. Singh, Image encryption and decryption using blowfish algorithm in MATLAB, *International Journal of Scientific & Engineering Research*, **4**, 7, 150-154 (2013).

[20] N. Singhal, J. Raina, Comparative analysis of AES and RC4 algorithms for better utilization, *International Journal of Computer Trends and Technology*, **2**, 6, 177-181 (2011).

[21] B. Hamouda, Comparative Study of Different Cryptographic Algorithms, *Journal of Information Security*, **11**, 138-148 (2020).

[22] T. Nie, C. Wang, X. Zhi, Performance Evaluation of DES and Blowfish Algorithms, *2010 International Conference on Biomedical Engineering and Computer Science*, **2020**, 1-4 (2010).

[23] T. Mahajan, S. Masih, *Enhancing Blowfish file encryption algorithm through parallel computing on GPU*, in Proc. 2015 International Conference on Computer, Communication and Control (IC4), 1-4, (2015). https://dpi.org/10.1109/IC4.2015.7375604.

[24] Konecný, J. Brozovský and V. Krivý, *Simulation Based Reliability Assessment Method using Parallel Computing*, in Proc. the First International Conference on Parallel, Distributed and Grid Computing for Engineering, 1-4, (2009). https://dpi.org/10.4203/ccp.90.38.

[25] W. Alisawi, Z. Oleiwi, W. Alawsi, A. Alfoudi, N. Hadi, Improvement of Classical Cipher Algorithm based on a New Model of Timed-Released Encryption, *International Journal of Applied Engineering Research*, **14**, 16, 3531-3536 (2019).

[26] Q. Zhang, L. Yang, Z. Chen, P. Li, High-order possibilistic c-means algorithms based on tensor decompositions for big data in IoT, *Information Fusion*, **39**, 72-80 (2018).

[27] S. Aljawarneh, M. Bani Yassein, W. Talafha, A multithreaded programming approach for multimedia big data: encryption system, *Multimedia Tools and Applications*, **77**, 9, 10997-11016 (2022). https://doi.org/10.1007/s11042-017-4873-9.

[28] Kondo, S. Tabu, Investigating the Efficiency of Secret Key Encryption Algorithms with similar key length and block size, *International Journal of Digital Information and Wireless Communications*, **10**, 2, 29-35 (2020).

[29] A. Shetty, K. Shravya, K. Krithika, A review on asymmetric cryptography–RSA and Elgamal algorithm, *International Journal of Innovative Research in Computer and Communication Engineering*, **2**, 5, 98-105 (2014).

[30] K. Valenza, Y. Heucheun, L.Mariel, A. Tiedeu, G. Kom, A Robust and Fast Image Encryption Scheme Based on a Mixing Technique, *Security and Communication Networks*, **2021**, 1-17 (2021). https://doi.org/10.1155/2021/6615708.

[31] Z. Hua, Y. Zhou, H. Huang, Chaotic system, Chaos-based encryption, Cryptography, Image privacy, Image encryption, Security analysis, *Information Sciences*, **480**, 403-419 (2019). https://doi.org/10.1016/j.ins.2018.12.048.

[32] M. Asgari-Chenaghlu, M.Balafar, M. Feizi-Derakhshi, A novel image encryption algorithm based on polynomial combination of chaotic maps and dynamic function generation, *Signal Processing*, **157**, 1-13 (2019). https://doi.org/10.1016/j.sigpro.2018.11.010.

[33] H. Wen, S. Yu, J. Lu, Breaking an Image Encryption Algorithm Based on DNA Encoding and Spatiotemporal Chaos, *Entropy*, **21**, 3, 246 (2019). https://doi.org/10.3390/e21030246.

[34] X. Xhang, X. Wang, Multiple-image encryption algorithm based on DNA encoding and chaotic system, *Multimedia Tools and Applications*, **78**, 6, 7841-7869 (2019). https://doi.org/10.1007/s11042-018-6496-1.

[35] M. Abu-Faraj A. Al-Hyari, K. Aldebei, Z. Alqadi, B. Al-Ahmad, Rotation Left Digits to Enhance the Security Level of Message Blocks Cryptography, *IEEE Access*, **2022**, 69388-69397 (2022). https://doi.org/10.1109/ACCESS.2022.3187317.

[36] M. Abu-Faraj Z. Alqadi, M. Zubi, Creating color image features based on morphology image processing, *Traitement du Signal*, **39**, 3, 797-803 (2022). https://doi.org/10.18280/ts.390304.

Mua'ad M. Abu-Faraj received the B.Eng. degree in Computer Engineering from Mu'tah University, Jordan, in 2004, the M.Sc. degree in Computer and Network Engineering from Sheffield Hallam University, UK, in 2005, and the M.Sc. and Ph.D. degrees in Computer Science and Engineering from the University of Connecticut, Storrs, USA, in 2012. He is, at present, an Associate Professor at The University of Jordan, Jordan. His research interests include computer architecture, reconfigurable hardware, image processing, cryptography, and wireless networking. Dr. Abu-Faraj is a member of the IEEE, and JEA (Jordan Engineers Association).

Abeer Al-Hyari Abeer Al-Hayri is an assistant professor at the electrical engineering department at Al-Balqa Applied University, As Salt, Jordan. Abeer's research involves cryptography, in addition to the application of machine learning, deep learning, recurrent neural networks to problems in FPGA CAD. Abeer received her Ph.D. degree in Computer Engineering from University of Guelph, Guelph, Canada.

Bilal Al-Ahmad received B.Sc. degree in computer information systems from Jordan University of Science Technology, Jordan, in 2006, M.Sc degree in computer information systems from Yarmouk University, Jordan,in 2009, PhD in software engineering from North Dakota State University, USA, in 2015. Currently, he is an assistant professor in Computer Information Systems department at The University of Jordan, Aqaba branch. His research interests include requirements engineering, software testing, software design, machine learning, and computer networks.

Ziad Alqadi received the B.E., M. E., and Dr. Eng. degrees from Kiev Polytechnic Institute. in 1980, 1983, and 1986, respectively. After working as, a researcher from 1986, an assistant professor from 1991 in the department of Electrical Engineering, Amman Applied College, and an Associate Professor from 1996 in the Faculty of Engineering Technology, he has been a professor at Albalqa Applied. since 2010. His research interest includes signal processing, image processing, data security and parallel processing.

Basel J. Ali earned a Bachelor of Commerce from Aligarh Muslim University, India, in 2001, a Master of Commerce -Accounting from Jai Narain Vays University, India, in 2012, and a Ph.D. in Accounting from University Malaysia Perlis, Malaysia, in 2017. He is currently an Assistant Professor at Applied Science University in Bahrain. His research interests include digital accounting, artificial intelligence in accounting, AIS, and digital accounting.

Abdullah Alhaj received B.Sc. and M.Sc degree in computer engineering from Lviv polytechnic institute - USSR, in 1988, PhD in Computer Science from Bradford University UK, in 2008. Currently, he is an associate professor in the Information Technology department at The University of Jordan, Aqaba branch. His research interests include computer architecture, networks, IT security, machine learning, and AI.