UDC 082
DOI: 10.15587/1729-4061.2021.225508

IMPLEMENTATION OF MODIFIED GSO BASED MAGIC CUBE KEYS GENERATION IN CRYPTOGRAPHY

Alaa Noori Mazher
Assistant Professor
Department of Computer Science
University of Technology
Baghdad, Iraq
E-mail: 110027@uotechnology.edu.iq

Jumana Waleed
PhD, Assistant Professor
Department of Computer Science
College of Science
University of Diyala
Baquba, Diyala, Iraq
E-mail: jumanawaleed@sciences.uodiyala.edu.iq

Over the last few decades, tremendous and exponential expansion in digital contents together with their applications has emerged. The Internet represents the essential leading factor for this expansion, which provides low-cost communication tools worldwide. However, the main drawback of the Internet is related to security problems. In order to provide secure communication, enormous efforts have been spent in the cryptographic field. Recently, cryptographic algorithms have become essential for increasing information safety. However, these algorithms require random keys and can be regarded as compromised when the random keys are cracked via the attackers. Therefore, it is substantial that the generation of keys should be random and hard to crack. In this paper, this is guaranteed via one of the most efficient nature-inspired algorithms emerged by inspiring the movements of stars, galaxies, and galaxy superclusters in the cosmos that can be utilized with a mathematical model (magic cube) for generating hardly cracking random number keys. In the proposed cryptographic system, the Modified Galactic Swarm Optimization (GSO) algorithm has been utilized in which every row and column of magic cube faces are randomly rotated until reaching the optimal face, and the optimal random elements are selected as optimal key from the optimal face. The generated optimized magic cube keys are used with several versions of RC6 algorithms to encrypt various secret texts. Furthermore, these generated keys are also used for encrypting images using the logical XOR operation. The obtained results of NIST tests proved that the generated keys are random and uncorrelated. Moreover, the security of the proposed cryptographic system was proved

Keywords: modified galactic swarm optimization (GSO), magic cube, key generation, cryptography

Received date 30.11.2020
Accepted date 19.01.2021
Published date 26.02.2021

1. Introduction

In the modern information world, the utilization of safe data transmitting protocols and cryptographic systems assists in protecting against several attacks on information security [1]. Cryptography represents a scheme in which the information is converted into a non-understood or unreadable style and the encryption and decryption processes are applied using programming algorithms over a digital field [2]. There are two fundamental kinds of cryptographic systems that are based on the utilization of keys; symmetric key and asymmetric key. In the symmetric key cryptographic systems, the sender and receiver share the same secret key to encrypt and decrypt messages. While the asymmetric key cryptographic systems require a pair of keys, the public key of the sender is utilized for encrypting messages, and the private key of the receiver is utilized for decrypting the ciphertext. The data security is based on the complexity and secrecy of the keys. In wide computer networks, compared with symmetric key cryptographic systems, asymmetric key cryptographic systems consume more time and storage space, since both the sender and receiver sides require to store a large number of keys, and utilize a larger amount of computation for encrypting and decrypting messages using different keys [3].

Several metaheuristic algorithms have emerged to search for the optimal potential solution in an acceptable time [4]. Metaheuristic algorithms are created by inspiring various disciplines such as physics, sociology, and biology. Nowadays, the most effective metaheuristic algorithms are inspired by the swarm behavior of biological systems, natural phenomena, and animals [5].

In cryptography, in order to secure the content, a lot of researches concentrate on making the problem of inference very hard. Therefore, metaheuristic algorithms can be utilized for generating random uncorrelated keys in cryptography [6, 7].

2. Literature review and problem statement

Cryptographic systems are needed for securing the transmission of secret information during communication. Day after day, the significance of security is increased owing to the rise of e-commerce and the processing of online transactions. In addition to the algorithms of encryption and decryption, the characteristic of the randomness for the generated key sequences proves the strength of the symmetric key cryptographic systems [7].

The magic cube can be regarded as a cube toy that is separated into distinct sub cubes. These sub cubes can be scrambled using the rotation, which is performed based on specific rules to yield completely different sub cubes. Hence, the concept of the magic cube has been utilized for generating random secret keys to be used for different cryptography and information hiding applications. Most of the researchers used the principle of rotation for shifting every row and column of the faces in the magic cube.

In [8], the transformation of a magic cube was utilized for shuffling the locations of the pixels in an image and modifying the values of the pixels using pseudo-random sequences generated via chaotic maps. In this image cryptography system, the obtained results illustrate that the proposed system is secure and effective enough, and the encrypted images cannot be recognized. While in [9], a system of image encryption in the frequency domain based on rotating the magic cube was proposed. In this system, firstly, the discrete fractional Fourier transform is used for encrypting the image. Secondly, an algorithm of image scrambling using the magic cube rotation and chaotic maps are applied for getting better results, compared to only utilizing the transform domain.

In [10], a system of keys generation and cryptography depending on combining magic cubes hybridization and rotation was presented. There are two processes of rotations implemented in this system. Firstly, generating the key by rotating the magic cube. Secondly, generating the ciphertext by rotating the original text. The obtained results show that the proposed system was secure to some cryptanalysis. In [11], an image cryptography system depending on the 3D magic cube was presented. This proposed system starts with RC6 to separately encrypt multi-images. The encrypted images are encrypted again with the 3D Rubik's cube, in other words, these encrypted images are utilized as faces for the Rubik's cube. The obtained results show that the proposed system is secure and efficient. However, these systems require to exhibit more robustness.

In [12], a system was proposed in which the magic cube is utilized as a 3-dimensional reference in hiding information into grayscale images. The secret information is transformed into the spatial coordinates and the least significant bits of the cover image are substituted regarding these coordinates. The utilization of the magic cube works on only embedding the secret bits via substitution without adding or decreasing the pixel's value, and this leads to avoiding the problem of overflow. While, in [13], a hybrid of the cryptographic and information hiding system was proposed. This hybrid system includes, firstly, generating a secret random key based on a magic cube and chaotic maps, secondly, the generated key is utilized for encrypting an image. Thirdly, the audio cover is utilized for embedding the encrypted image. In this system, in order to generate the random key, the number of iteration and the initial values of chaotic maps are firstly entered, and the obtained numbers are stored in an array. This array is used for filling the magic cube faces and used also for rotating the magic cube and selecting the key. Another information hiding system based on the magic cube key generation was proposed in [14]. In this system, firstly the magic cube based key is generated based on a magic matrix, then, the cover image is separated into non-overlapped blocks. After that, every nine bits from the secret medical data are embedded in each block based on the generated keys. These systems require carrying out more investigations to design better magic cubes to improve image quality.

A few of the above-mentioned related systems successfully elapsed most of the NIST tests, but these systems neglected the principle of finding optimal randomness for the generated magic cube based keys, thus, this important requirement should be achieved. In this paper, this is guaranteed by the Modified GSO algorithm, which is used for selecting an optimal random magic cube based keys.

3. The aim and objectives of the study

The aim of the proposed system is to generate optimal, random, and uncorrelated keys. The generated key is utilized in the applications of cryptography.

To achieve the aim, the following objectives were set:

– to utilize the magic cube to provide a computational hardness for finding the solution;

– to utilize the Modified GSO algorithm for selecting an optimal random magic cube based key. This optimization algorithm works in two directions; the first one is to find the optimal face after applying many rows and columns rotations; the second direction is to find the optimal elements within the optimal face, these elements represent the optimal random key;

– to encrypt the secret plaintext messages using the Rivest Cipher 6 (RC6) algorithm and encrypt the secret images using the XOR operation depending on the generated keys.

4. Proposed cryptographic system

This section presents the proposed cryptography system. The proposed system relates between a new metaheuristic algorithm, magic cube, and cryptography. The general diagram of the proposed system is shown in Fig. 1.

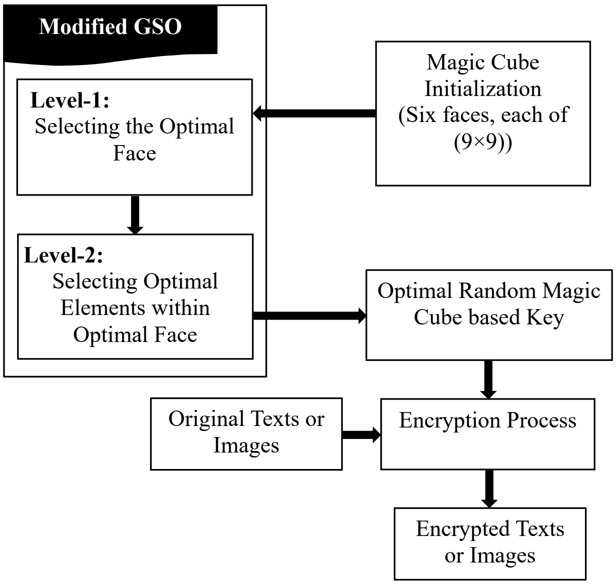
Modified GSO
Level-1: Selecting the Optimal Face
Level-2: Selecting Optimal Elements within Optimal Face
Magic Cube Initialization (Six faces, each of (9×9))
Optimal Random Magic Cube based Key
Original Texts or Images
Encryption Process
Encrypted Texts or Images

Fig. 1. General diagram of the proposed system

The proposed system is based on an optimal magic cube key generation, and it includes several stages. Firstly, the magic cube is constructed and then the modified GSO algorithm is utilized in order to generate optimal random keys, after that, depending on the generated keys, the secret plaintext messages are encrypted by using the RC6 algorithm and secret images are encrypted by using XOR operation.

4. 1. Magic cube generation

In 1974, Ernő Rubik invented the magic cube that originally contains six faces and each face includes nine stickers. Each face can be rotated clockwise or counterclockwise. The magic cube represents a three-dimensional integer's matrices (faces), ordered in the pattern of m×m×m. The essential characteristic of the magic cube is that the summation of the numbers in each row, each column, and main diagonal is equal to a constant number named magic cube constant, indicated by $C_3(m)$ given as follows:

$$C_3(m) = \frac{m\left(m^2+1\right)}{2}. \tag{1}$$

The concept of rotating the magic cube can be utilized for generating random secret keys to be used for different applications of cryptography and steganography. The larger number of rotations leads to the higher complexity of the locations on the magic cube and consequently leads to obtaining random keys.

In the process of magic cube generation, the number of faces and the number of elements in each face are based on the required key length of the encryption process. In order to generate 128-bit, the magic cube includes 6 faces, each face with 9×9. The initialization of each face in the magic cube is done according to equation (6), where the summation of the numbers in each row, each column, and main diagonal is equal to 369, see Fig. 2.

45	34	23	12	1	80	69	85	47
46	44	33	22	11	9	79	68	57
56	54	43	32	21	10	8	78	67
66	55	53	42	31	20	18	7	77
76	65	63	52	41	30	19	17	6
5	75	64	62	51	40	29	27	16
15	4	74	72	61	50	39	28	26
25	14	3	73	71	60	49	38	36
35	24	13	2	81	70	59	48	37

Fig. 2. Example of random selection exists in all six surfaces of magic cube

The random values shown in Fig. 2 exist on all the faces of magic cube as an example of random selection of numbers.

4. 2. Modified GSO algorithm based magic cube key generation

The behavior of stars within galaxies and galaxies in superclusters under the gravity influence was used in the inspiration of the GSO algorithm. GSO works on taking the advantages of particle swarm optimization (PSO) with the use of flexible multiple cycles of exploration and exploitation levels for finding optimal solutions. It can be applied in various problems of real life owing to its efficiency and simple implementation [15].

In the GSO algorithm, the stars' movement within a galaxy and the galaxies' movement are emulated based on the following rules. Firstly, within each galaxy, stars (individuals) are attracted to better ones (optimal solutions). This process of attraction is applied by utilizing the algorithm of PSO. Secondly, to all galaxies, the global bests are selected to be treated as a super swarm. The PSO is also utilized to perform the motion of particles in the super swarm.

The GSO algorithm includes several significant controlling parameters to be utilized in each level. The population or swarm (supercluster of galaxies) consists of N partitions (galaxies of stars) named sub-swarms $S_i \subset S : i = 1,2,\ldots,N$, which include n elements or stars $S_j^{(i)} \subset S_i : j = 1,2,\ldots,n$, where $S_i \cap S_j = \varnothing, \forall i \neq j$ and $\left(S_j^{(i)} \in R^{dim}\right)$. The initialization of each star is generated randomly inside the search space $[s_{\text{Minimum}}, s_{\text{Maximum}}]^{dim}$.

At level-1 of the GSO algorithm, the algorithm of PSO is performed independently to every sub swarm, thus, PSO should be run N times. Each sub swarm holds an associated global best $gb^{(i)}$, and this $gb^{(i)}$ can be updated when one of its personal bests $pb_j^{(i)}$ takes a value of minimal function than $gb^{(i)}$, $O\left(pb_j^{(i)}\right) < O\left(gb^{(i)}\right)$, where O is the objective function. Stars in every sub swarm are attracted randomly to the local minimum (optimal solution) reached via that certain sub swarm. The motion of a sub swarm is independent without influence on the other sub swarms, hence, giving potential comprehensive and unaffected search. In order to obtain the whole interest of exploration capability of multi-sub swarms, galactic best gb is updated when any global best $gb^{(i)}$ supposes a value of the preferable function, $O\left(gb^{(i)}\right) < O(gb)$. GSO can keep a record for its suitable solution through updating gb.

Every sub swarm explores independently the search space on its own. The iteration starts by calculating the velocity $v_j^{(i)}$ and location of stars. The following expressions explain the updating of the star's velocity and location:

$$v_j^{(i)} \leftarrow w_1 v_j^{(i)} + cc_1 rand_1 \left(pb_j^{(i)} - S_j^{(i)}\right) + \\ + cc_2 rand_2 \left(gb^{(i)} - S_j^{(i)}\right), \tag{2}$$

$$S_j^{(i)} \leftarrow S_j^{(i)} + v_j^{(i)}, \tag{3}$$

where w_1 is the inertia weight, cc_1 and cc_2 are the coefficients of acceleration for PSO, $rand_1$ and $rand_2$ represent random numbers within –1, and 1.

At level-2 of the GSO algorithm, the global bests are involved to compose superclusters. In the same way, the new super swarm is generated through the aggregation of the global bests from sub swarms.

$$l^{(i)} \in l : i = 1,\ldots,N. \tag{4}$$

where; $l^{(i)}$ is the location of super swarm i, N is the number of sub-swarms, $l^{(i)}=gb^{(i)}$. The vectors of velocity and location are updated as follows:

$$v^{(i)} \leftarrow w_2 v^{(i)} + cc_3 rand_3 \left(pb^{(i)} - l^{(i)}\right) + \\ + cc_4 rand_4 \left(gb - l^{(i)}\right), \tag{5}$$

$$l^{(i)} \leftarrow l^{(i)} + v^{(i)}, \tag{6}$$

where w_2 is the inertia weight, cc_3 and cc_4 are the coefficients of acceleration for PSO, $rand_3$ and $rand_4$ represent random numbers within –1, and 1.

At this level, gb denotes as the global best sample and this gb will not be updated until the search specified the preferable point. As the super swarm focuses on the global bests from sub swarm, it is capable of enhancing the exploitation. The location of galactic best after the last epoch gb and its value of evaluation $O(gb)$ are returning as minimum cost and location, respectively by the algorithm.

In the original GSO algorithm, the first level starts with finding the optimal stars in all the galaxies, after that, in the second level, the optimal galaxy is selected. This strategy will lead to losing a lot of time. In this proposed system, we will modify this strategy to be suitable to cope with our problem and prevent time wasting by selecting the optimal galaxy then selecting the optimal stars where the galaxies represent the faces of the magic cube, and the stars represent the elements in each face. Fig. 3 shows an example of this modified strategy. The modified GSO strategy based magic cube works as follows:

– modified GSO parameter initialization;

– at Level-One, in order to decrease the opportunity of pattern prediction by the cryptanalysts, thus, randomly rotating every row and column of magic cube faces, and selecting the optimal face (galaxy) according to equations (5) and (6);

– at Level-Two, within the optimally selected face, the PSO works on finding the optimal elements, which represent the optimal key by using equations (2) and (3).

4. 3. Processes of cryptography

In this stage, the obtained optimal random key bits are used with several versions of RC6 algorithms to encrypt and decrypt various plain texts.

RC6 algorithm represents a symmetric key block cipher. In general, it holds several variable parameters like block size, number of rounds, and key size. Particularly, the RC6 encryption algorithm is designed as RC6 (*d*, *n*, *l*), where *d* represents the size of the word (each block includes four words), *n* represents the number of rounds and *l* represents the length of the secret key in bytes. Besides the standardized RC6, there are various versions of RC6 that are capable of working on the block size equal to 128 bits using 28 bits, 192 bits, ..., 2,040 bits of key sizes.

In the RC6encryption algorithm, firstly, a process of expanding the secret key to an array of 2*n*+4 secret *d*-bit words S_k in accordance with the algorithm of key scheduling is carried out. Suppose (*W*, *X*, *Y*, *Z*) represent the plaintext involving four *d*-bit words. Point out that a d-bit word is identically described as a Modular Arithmetic (integer modulo 2^d). After that, the encryption algorithm is specified as explained in Fig. 3.

```
$X \leftarrow X + S_0 \bmod 2^d$
$Z \leftarrow Z + S_1 \bmod 2^d$
For $k = 1$ to $n$ do
$W \leftarrow ((W \oplus f(X)) \ll f(Z)) + S_{2i} \bmod 2^d$
$Y \leftarrow ((Y \oplus f(Y)) \ll f(X)) + S_{2i+1} \bmod 2^d$
$(W, X, Y, Z) \leftarrow (X, Y, Z, W)$
$W \leftarrow W + S_{2n+2} \bmod 2^d$
 $Y \leftarrow Y + S_{2n+3} \bmod 2^d$
```

Fig. 3. RC6 encryption algorithm

Where ⊕ is the operation of bit-wise Exclusive OR, «A<<B» represents the rotation of A to the left via the $\log_2 d$ least significant bits of B, and *f* represents the function that holds the role of a pseudorandom generator given by:

$$f(a)=\left(a(2a+1)\bmod 2^d\right)<<\log_2 d. \qquad (7)$$

The cipher text represents (*W*, *X*, *Y*, *Z*) and every round *n* includes precisely two sub-keys S_k.

Furthermore, in this stage, the optimal generated keys are also used for encrypting grayscale and color images using the logical XOR operation.

5. Experimental results

In this section, several experiments are presented for proving the validity of the proposed optimal magic cube based key generation system. Fig. 4 shows an example of the faces after applying the Modified GSO, and the optimal face.

33	11	46	22	44	79	68	57	9
74	61	15	72	4	39	28	26	50
23	1	45	12	34	69	58	47	80
64	51	5	62	75	29	27	16	40
63	41	76	52	65	19	17	6	30
3	71	25	73	14	49	38	36	60
53	31	66	42	55	18	7	77	20
43	21	56	32	54	8	78	67	10
13	81	35	2	24	59	48	37	70

a

47	45	58	34	1	12	23	69	80
67	56	78	54	21	32	43	8	10
36	25	38	14	71	73	3	49	60
16	5	27	75	51	62	64	29	40
57	46	68	44	11	22	33	79	9
37	35	48	24	81	2	13	59	70
26	15	28	4	61	72	74	39	50
77	66	7	55	31	42	53	18	20
6	76	17	65	41	52	63	19	30

b

18	20	42	31	55	66	77	7	53
8	10	32	21	54	56	67	78	43
19	30	52	41	65	76	6	17	63
29	40	62	51	75	5	16	27	64
69	80	12	1	34	45	47	58	23
79	9	22	11	44	46	57	68	33
39	50	72	61	4	15	26	28	74
59	70	2	81	24	35	37	48	13
49	60	73	71	14	25	36	38	3

c

68	22	33	9	79	46	57	44	11
17	52	63	30	19	76	6	65	41
7	42	53	20	18	66	77	55	31
58	12	23	80	69	45	47	34	1
38	73	3	60	49	25	36	14	71
27	62	64	40	29	5	16	75	51
48	2	13	70	59	35	37	24	81
78	32	43	10	8	56	67	54	21
28	72	74	50	39	15	26	4	61

d

26	39	50	4	61	72	74	15	28
6	19	30	65	41	52	63	76	17
16	29	40	75	51	62	64	5	27
47	69	80	34	1	12	23	45	58
67	8	10	54	21	32	43	56	78
77	18	20	55	31	42	53	66	7
57	79	9	44	11	22	33	46	68
37	59	70	24	81	2	13	35	48
36	49	60	14	71	73	3	25	38

e

12	45	69	47	58	1	80	23	34
2	35	59	37	48	81	70	13	24
72	15	39	26	28	61	50	74	4
62	5	29	16	27	51	40	64	75
52	76	19	6	17	41	30	63	65
32	56	8	67	78	21	10	43	54
42	66	18	77	7	31	20	53	55
22	46	79	57	68	11	9	33	44
73	25	49	36	38	71	60	3	14

f

68	22	33	9	79	46	57	44	11
17	52	63	30	19	76	6	65	41
7	42	53	20	18	66	77	55	31
58	12	23	80	69	45	47	34	1
38	73	3	60	49	25	36	14	71
27	62	64	40	29	5	16	75	51
48	2	13	70	59	35	37	24	81
78	32	43	10	8	56	67	54	21
28	72	74	50	39	15	26	4	61

g

Fig. 4. Example: *a, b, c, d, e, f* – six random faces after applying the GSO; *g* – optimal face

After applying the modified GSO strategy, the optimal face (optimal solution) is obtained when performing many rotations for the magic cube, which represent an optimal objective function. Then, the optimal 16 elements (local stars) are selected from 81 elements. These elements are converted to 128 bits. For the above example, the optimal elements in the optimal selected face are «24, 76, 80, 33, 20, 62, 6, 1, 45, 15, 5, 27, 47, 79, 40, 77», and the generated binary random key is «0001100000100110001010000001000010001010000011111000000110000000001001011010000111100000101000110110010111101001111001010000 1001101».

The Suite of NIST tests are utilized for checking the randomness of the generated cryptographic key, these tests represent ten statistical tests. Table 1 explains the obtained results of NIST tests for the generated binary key bits.

Table 1

Results of NIST tests for the generated optimal magic cube based key

NIST Tests	P_Value>0.01	States
Block Frequency Testd	0.6022	Success
Cumulative Sums Test	0.8110	Success
DFT Test	0.5330	Success
Run Test	0.3299	Success
Frequency Test	0.899	Success
Longest Run of One's Test	0.9177	Success
Non Overlapping Test	0.7889	Success
Ndtr Test	0.9943	Success
Evaluate Bit Stream	0.6829	Success
Rank Test	0.5001	Success

The generated optimal random keys are utilized in texts and images cryptography. In the application of texts cryptography, the secrecy of ciphers can be computed in terms of the key equivocation (conditional entropy of key given cipher). Table 2 shows the plain texts, the obtained cipher texts of variable lengths using RC6 with the optimal generated key, and the average security for obtained cipher texts using the optimal generated key and using randomly selected key from the magic cube.

The RC6 algorithm has been improved using the generated key regardless of the size of plain text.

In the application of images cryptography, several images of size (256×256) have been utilized to be encrypted using XOR operation. Fig. 5 shows the obtained encrypted images with their histograms.

Table 2

Plain texts and obtained cipher texts of variable lengths using RC6

Plain Size	Plain Text	Cipher Text in Hexadecimal	Cipher Text in Unicode	Average Security Using Optimal Key	Average Security Using Random Selected Key
32 bit	'iraq'	['20';'B8';'38';'5C']	,8\	0.112103	0.097785
64 bit	'Computer'	['0A';'A5';'34';'5D'; '89';'80';'FA';'DA']	¥4]M	0.125864	0.112532

128 bit	'Computer Science'	['0A';'A5';'34';'5D';
		'89';'80';'FA';'DA';
		'4D';'57';'66';'E5';
		'52';'4A';'F9';'93']
			¥4]MWfRJù	0.129075	0.104257

256 bit	'Computer Science, Data Security '	['0A';'A5';'34';'5D';
		'89';'80';'FA';'DA';
		'4D';'57';'66';'E5';
		'52';'4A';'F9';'93';
		'72';'D0';'47';'19';
		'62';'62';'46';'B6';
		'8E';'4E';'24';'79';
		'20';'7C';'5F';'DC']
			¥4]MWfR-JùrGbb-F¶N$y |_	0.298383	0.185486

The histogram describes the intensity distribution of the image. It represents a visual tool that is utilized for seeking the ultimate relationship between the original and encrypted distributions of images. So, the histogram of encrypted images should be flat and uniformly distributed. Here, the obtained histograms of encrypted images are equally distributed.

Besides the utilization of peak signal to noise ratio (PSNR) as an evaluation criterion for image quality, it can be utilized for finding the closeness between the encrypted and the original images. Additionally, the Correlation Coefficient can be utilized for the same purposes. Table 3 shows the results of PSNR and Cross Correlation for encrypted images using the optimal generated key and using a randomly selected key from the magic cube.

Table 3

Results of PSNR and Correlation Coefficient for encrypted images

Encrypted Images	With Optimal Key		With Random Selected Key	
	PSNR	Correlation Coefficient	PSNR	Correlation Coefficient
Lena	7.438152	0.460107	12.175318	0.610376
Trees	6.689454	0.585492	10.671716	0.658691
Peppers	7.597426	0.483693	11.185189	0.596993
Baboon	8.727593	0.692456	13.670795	0.828673

Whenever the values of PSNR are lower, the higher the difference between the original and the encrypted images, resulting in a more secure image cryptography system. And in the criterion of Correlation Coefficient, a correlation of –1.0 shows a perfect negative correlation, while a correlation of 1.0 shows a perfect positive correlation.

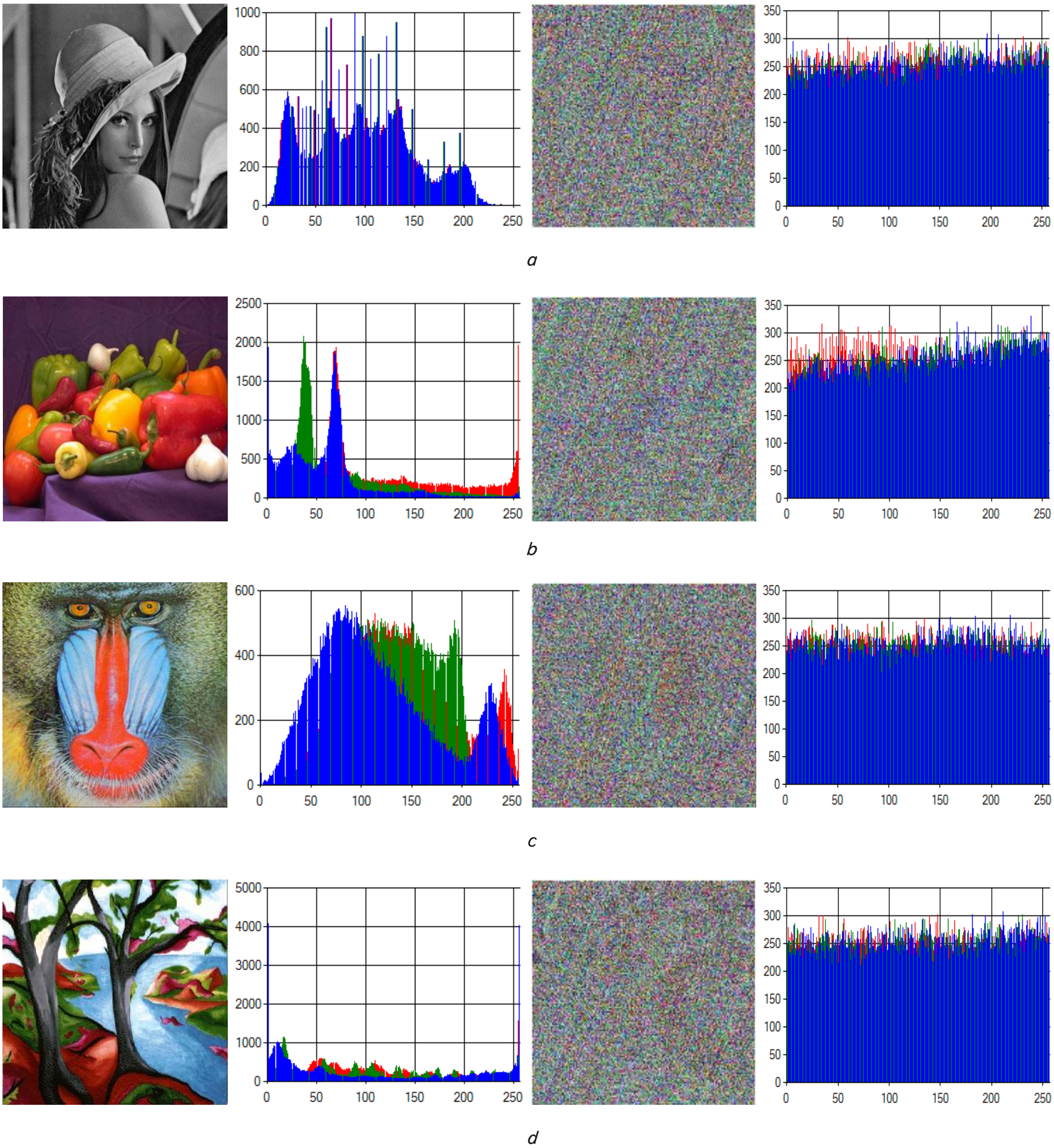

Fig. 5. Original images with their histograms, and encrypted images with their histograms: *a* – Lena image; *b* – peppers image; *c* – baboon image; *d* – trees image

6. Discussion of the proposed system

The magic cube-based issue provides a computational hardness for obtaining the solution, subsequently, attackers are facing difficulty to reach a solution for the scrambled magic cube since it can find more possibilities in the mathematical functions form like factorial and exponential. In this paper, the Modified GSO algorithm has been exploited for randomly rotating the faces of the magic cube to find the optimal face, and the optimal random key is obtained from the optimal face. The obtained results of NIST tests demonstrate that the generated random keys successfully passed all NIST tests. These generated keys are utilized in texts and images cryptography.

The obtained results of average security from the experiments point out that the proposed text cryptography system achieves high results.

In order to evaluate the proposed image cryptography system, the histogram of both original and encrypted images was calculated. As previously seen from the histograms in Fig. 5, the histograms of original images have a tilted structure analogous to the original images, however, the histograms of the encrypted images are uniformly distributed and flat. Consequently, statistical attacks utilizing the information of histograms are impossible. Moreover, the values of PSNR of Lena, Trees, Peppers, and Baboon images encryption are tested. The obtained results of small PSNR values demonstrate that the original and encrypted images have considerable differences. Another measurement of evaluation is the correlation coefficient. The experimental results demonstrate that the encrypted images have very low correlations. Therefore, the proposed image cryptography system is efficient and secure. In this paper, the optimization algorithm was implemented only on a magic cube. In future works, we can develop a new optimization algorithm based on the magic rectangle with m dimensions, and present a symmetric cipher based on this optimized magic rectangle key.

7. Conclusions

1. The mathematical phenomenon (magic cube) can be successfully created as an extension to the mathematical magic square.

2. The obtained results of NIST tests explain that the generated optimized magic cube based key is random with no correlation.

3. The RC6 algorithm has been improved using the generated key regardless of the size of plain text. Also, the RC6 algorithm has been improved by utilizing the optimal keys of the magic cube regardless of the key length. Furthermore, the image cryptography algorithm based on the generated keys has higher security and the obtained results of average PSNR and Correlation Coefficient for the encrypted images are 7.613156, and 0.5554, respectively.

References

1. Ruzhentsev, V., Onishchenko, Y. (2017). Development of the approach to proving the security of block ciphers to impossible differential attack. Eastern-European Journal of Enterprise Technologies, 4 (4 (88)), 28–33. doi: https://doi.org/10.15587/1729-4061.2017.108413
2. Mazhar, A. N., Naser, E. F. (2020). Hiding the Type of Skin Texture in Mice based on Fuzzy Clustering Technique. Baghdad Science Journal, 17 (3), 967–972. doi: https://doi.org/10.21123/bsj.2020.17.3(suppl.).0967
3. Indrasena Reddy, M., Siva Kumar, A. P., Subba Reddy, K. (2020). A secured cryptographic system based on DNA and a hybrid key generation approach. Biosystems, 197, 104207. doi: https://doi.org/10.1016/j.biosystems.2020.104207
4. Waleed, J., Jun, H. D., Hameed, S. (2015). An Optimized Digital Image Watermarking Technique Based on Cuckoo Search (CS). ICIC Express Letters. Part B, Applications: an international journal of research and surveys, 6 (10), 2629–2634.
5. Kaya, E., Uymaz, S. A., Kocer, B. (2018). Boosting galactic swarm optimization with ABC. International Journal of Machine Learning and Cybernetics, 10 (9), 2401–2419. doi: https://doi.org/10.1007/s13042-018-0878-6
6. Jaya Krishna, G., Ravi, V., Nagesh Bhattu, S. (2018). Key generation for plain text in stream cipher via bi-objective evolutionary computing. Applied Soft Computing, 70, 301–317. doi: https://doi.org/10.1016/j.asoc.2018.05.025
7. Sudeepa, K. B., Aithal, G., Rajinikanth, V., Satapathy, S. C. (2020). Genetic algorithm based key sequence generation for cipher system. Pattern Recognition Letters, 133, 341–348. doi: https://doi.org/10.1016/j.patrec.2020.03.015
8. Zhu, Z., Wang, C., Chai, H., Yu, H. (2011). A Chaotic Image Encryption Scheme Based on Magic Cube Transformation. 2011 Fourth International Workshop on Chaos-Fractals Theories and Applications. doi: https://doi.org/10.1109/iwcfta.2011.75
9. Feng, X., Tian, X., Xia, S. (2011). A novel image encryption algorithm based on fractional fourier transform and magic cube rotation. 2011 4th International Congress on Image and Signal Processing. doi: https://doi.org/10.1109/cisp.2011.6100319
10. Rajavel, D., Shantharajah, S. P. (2012). Cubical key generation and encryption algorithm based on hybrid cube's rotation. International Conference on Pattern Recognition, Informatics and Medical Engineering (PRIME-2012). doi: https://doi.org/10.1109/icprime.2012.6208340
11. Helmy, M., El-Rabaie, E.-S. M., Eldokany, I. M., El-Samie, F. E. A. (2017). 3-D Image Encryption Based on Rubik's Cube and RC6 Algorithm. 3D Research, 8 (4). doi: https://doi.org/10.1007/s13319-017-0145-8
12. Wu, Q., Zhu, C., Li, J.-J., Chang, C.-C., Wang, Z.-H. (2016). A magic cube based information hiding scheme of large payload. Journal of Information Security and Applications, 26, 1–7. doi: https://doi.org/10.1016/j.jisa.2015.08.003
13. Redha, D. A., Mohsen, M. M. A. (2017). Multi-level Security Based on Dynamic Magic Cube and Chaotic Maps. Iraqi Journal of Information Technology, 7 (4), 106–127. doi: https://doi.org/10.34279/0923-007-004-009
14. Lee, C.-F., Shen, J.-J., Agrawal, S., Wang, Y.-X., Lee, Y.-H. (2020). Data Hiding Method Based on 3D Magic Cube. IEEE Access, 8, 39445–39453. doi: https://doi.org/10.1109/access.2020.2975385
15. Nguyen, B. M., Tran, T., Nguyen, T., Nguyen, G. (2020). Hybridization of Galactic Swarm and Evolution Whale Optimization for Global Search Problem. IEEE Access, 8, 74991–75010. doi: https://doi.org/10.1109/access.2020.2988717