Retina Image and Bat-Inspird Algorithm for Artificial Key Generation

Mohammed JasimRidha[1], Ali Hussein Fadil[2]

[1]Lecturer,College of Basic Education, Almustansiriayh University, [2]Lecturer, Presidency University, University of Diyala

Abstract

The process of key generation is utilized in different applications based on encryption techniques. The characteristics of the generated key should satisfy the required security as potential. The traditional techniques used for generating keys are based on various equations based on chaos maps or modulation for making these keys as one-time pads. Also, there are many techniques that utilized the block cipher or hash function for generating keys. In this paper, a retina image and bat-inspired algorithm are used for generating secure keys. Several processes have been applied to the input retina images like enhancement and edge detection processes. In the proposed technique, some parameters are adjusted for controlling the results and every output represents a bat solution and depending on the fitness function, the keys are generated. These generated keys are tested and evaluated using NIST tests. As a case study, some images are encrypted using the generated keys and the obtained encrypted images passed all the required security tests.

Keywords: *Retina Image, Bat-Inspired Algorithm (BIA), Keys generation, images encryption.*

Introduction

A stream cipher cryptography is widely utilized in the fields of communication and security. The attributes of this cryptography are based on the generation of the keys. Therefore, it is beneficial for generating accurate keys with a high degree of randomness since it is very significant for cybersecurity[1]. Generally, there are two kinds of random number generators (RNGs); pseudorandom number generators (PRNGs) (or named deterministic RNGs), and true random number generators (TRNGs) (or named non-deterministic RNGs) [2]. Usually, PRNGs are based on the cryptography algorithms and input seeds, whilst the TRNGs are based on the physical hardware and are capable of directly generating random numbers. The entropy of PRNGs output sequences is coming from the seeds, whilst the TRNGs entropy is coming from the physical signals [3]. This means the PRNGs are only capable of decreasing the entropy or remaining identical [4]. If the seeds of PRNGs are controlled or known via an attacker, then it is possible to predict the PRNGs output sequences, and this case represents the low entropy secret leakage[5]. Therefore, generally, the PRNGs designers assume that the seeds have high entropy and the attacker is not capable of predicting the value of the seeds. Practically, there are lots of PRNGs applications that utilize multiple sources of entropy because more sources of entropy are mixed, therefore, the entropy is high, for instance, in the Android system, the seeds in Open SSL's PRNGs use nine sources. But, the output of these PRNGs remains predictable by an attacker[6]. Additionally, there are more instances such as the PRNGs output predictability in the Linux system[7] and PRNGs recoverability in the Brillo operation systems, and PRNGs vulnerability in the implementations of Java[8]. The PRNGs are deterministic algorithms, therefore, formally, the PRNGs security can be proven. The loss of entropy in PRNGs was analyzed via the assistance of CBMC[9] in[10], in which the static analysis was used to check the PRNGs program codes. The PRNGs are not capable of increasing the input seeds entropy, while they should capable of keeping it identical. Therefore, the loss of entropy refers to the PRNGs have several possible issues.

Keys Generation: In cryptography, the process of key generation works on generating keys. The key generator (keygen) is a program or device utilized for

generating keys. These keys are utilized for encrypting and decrypting different kinds of data. In modern cryptography systems, the generation of keys includes the algorithms of symmetric-key (utilize a single shared key), and the algorithms of public-key (utilize public & private keys)[11]. In the algorithms of symmetric-key (like DES, and AES), the keys should be secret to keep the secrecy of data. While, in the algorithms of public-key (like RSA), the public keys are provided for anybody (usually via a digital certificate). The sender works on encrypting data with the public key of the receiver, but, only the private key holder is capable of decrypting these data [12]. Because the algorithms of public-key are considerably slow compared with the algorithms of symmetric-key, modern systems like SSH and TLS utilize incorporation of public-key and symmetric-key algorithms, in which the recipient receives the sender's public key and encrypts a piece of data (generated using some data or symmetric key). The rest of the conversation utilizes the algorithm of symmetric-key which is typically faster for encryption. Computer cryptography utilizes the integers to the keys. In certain circumstances, the keys are generated randomly by utilizing PRNGs. PRNGs are computer algorithms works on producing random data under analysis. Generally, when the PRNGs utilize the system entropy for the seed data, then better results are produced because the initial conditions of the PRNGs become very tricky for guessing them by the attackers. One of the other ways of generating randomness is to use the information beyond the systems. The disk encryption software uses the movements of user mouse for generating exceptional seeds, Here, the users are worked on moving the mouse irregularly. Elsewhere, the keys are acquired deterministically utilizing passphrases and keys acquisition functions.

Bat-Inspired Algorithm: The bat-inspired algorithm (BIA) was founded depending on the process of bats echolocation. Within the process of echolocation, the bats will work on creating pulses that are alive from eight to ten milliseconds with a fixed frequency and identical wavelength. There are several bats features were prepared for developing BIA; Firstly, although with no vision, the bats are capable of sensing and estimating the distance between the food and the hindrances beyond them; Secondly, If the bats begin flying for finding the food, then they are related with the location, velocity, varying loudness, wavelength, and constant frequency; Thirdly, for changing the loudness values from a small constant to a higher positive values, different schemes are attributed [16].

The BIA was founded in 2010 by X. S. Yang[17]. The basic mechanism for BIA is in[18,19]. The initialization of the algorithm is started, at the moment $t = 0$, given the position of the bat, the number of bats, velocity, pulse frequency, pulse loudness, and pulse velocity. After that, the speed and position of the bat are updated at the time t based on the following equations:

$$V_a^{t+1}=V_a^t+(X_a^t - X') \times fr_a \quad ...(1)$$

$$X_a^{t+1}=X_a^t+V_a^t \quad ...(2)$$

$$fr_a = fr_{min} + (fr_{max} - fr_{min}) \times random \quad ...(3)$$

Where indicated the bat velocity and indicated the bat position at t, and x' is the optimal position at t. To achieve that goal, fr_a is the bat ultrasonic frequency, fr_{min} represents the smallest pulse frequency and fr_{max} represents the highest pulse frequency, and denotes the new bat position and denotes the new bat velocity at t+1. At initial, for every bat, the value of pulse frequency is given randomly that is uniformly selected from [fr_{max}, fr_{min}].

Generating a random number *random1*. When *random1*> pulse rate, then a random walk method is utilized to locally search about the bat for generating a new solution, as explained in the next equation:

$$X_{new} = X_{old} + \delta A^t \quad ...(4)$$

Where denotes a random solution selected from the current optimal solutions, is a random vector which is selected from [1, -1], and A^tdenotes the loudness.

Generating a random number *random2*. When *random2*> and the new solution fitness is optimal than the old one, then, the old solution is replaced with the new one and updated it as in the next equations:

$$r_a^t = r_0[1 - exp(-\gamma * t)] \quad ...(5)$$

$$A_a^t=\alpha \times A_a^{t-1} \quad ...(6)$$

Where is the pulse frequency of the bat at t, and r_0 denotes a constant, denotes the bat pulse intensity at t-1, denotes the bat pulse intensity at t, and are coefficients.

After that, finding the preferable fitness bat after updating and recording it. Checking if the highest number of iterations is achieved or the search accuracy is achieved. When it is achieved, the iteration is finished, and the optimum position of the bat with the optimum level of fitness are output.

The Proposed Key Generation System: In this proposed key generation system, a retina image and bat-inspired algorithm (BIA) are used for generating secure keys. There are several steps are applied to the input retina images like retina image enhancement and edge detection processes. In the proposed system, some parameters are adjusted for controlling the results and every output represents a bat solution and depending on the fitness function, the keys are generated. After that, these generated keys are utilized for encrypting some grayscale images. The structure of the proposed key generation system

A. Retina Image Enhancement Step: The enhancement of the retina images is the initial step in the proposed key generation system. In this preprocessing step, the input color retina image is separated into three channels; Red, Green, and blue, after that, the contrast of these channels is adjusted

B. The Step of Edges Detection: In this step, the enhanced retina image obtained from the former step is converted to a grayscale image, then, the Gaussian filter is utilized on it. After that, the process of extracting the weak and strong edges is done and placed in isolated images. The final retina image is obtained based on the results of canny edge detector which are obtained from the isolated images.

C. Position Initialization in BIA: The process of initializing the points of BIA is depending on Bresenham's Circle Drawing Algorithm and the detected edges from the former step. The circle is composed of eight equal Octets, therefore, only the coordinates are required to be found, octet-2 is taken, and "X and Y" will refer to the pixel. The points are selected from canny edges points that surrounded by two circles with different radius.

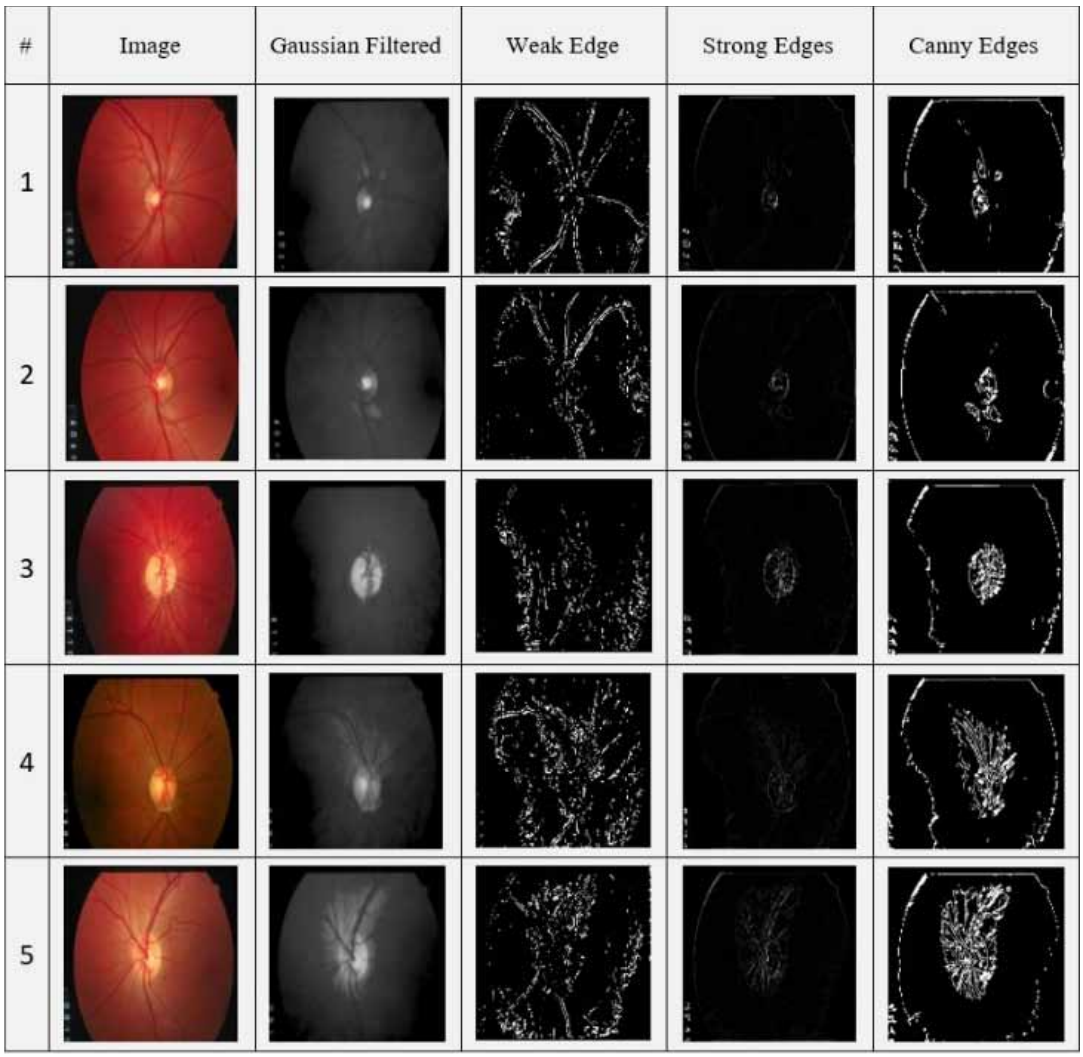

Figure 1: Some image samples from the dataset and their output images.

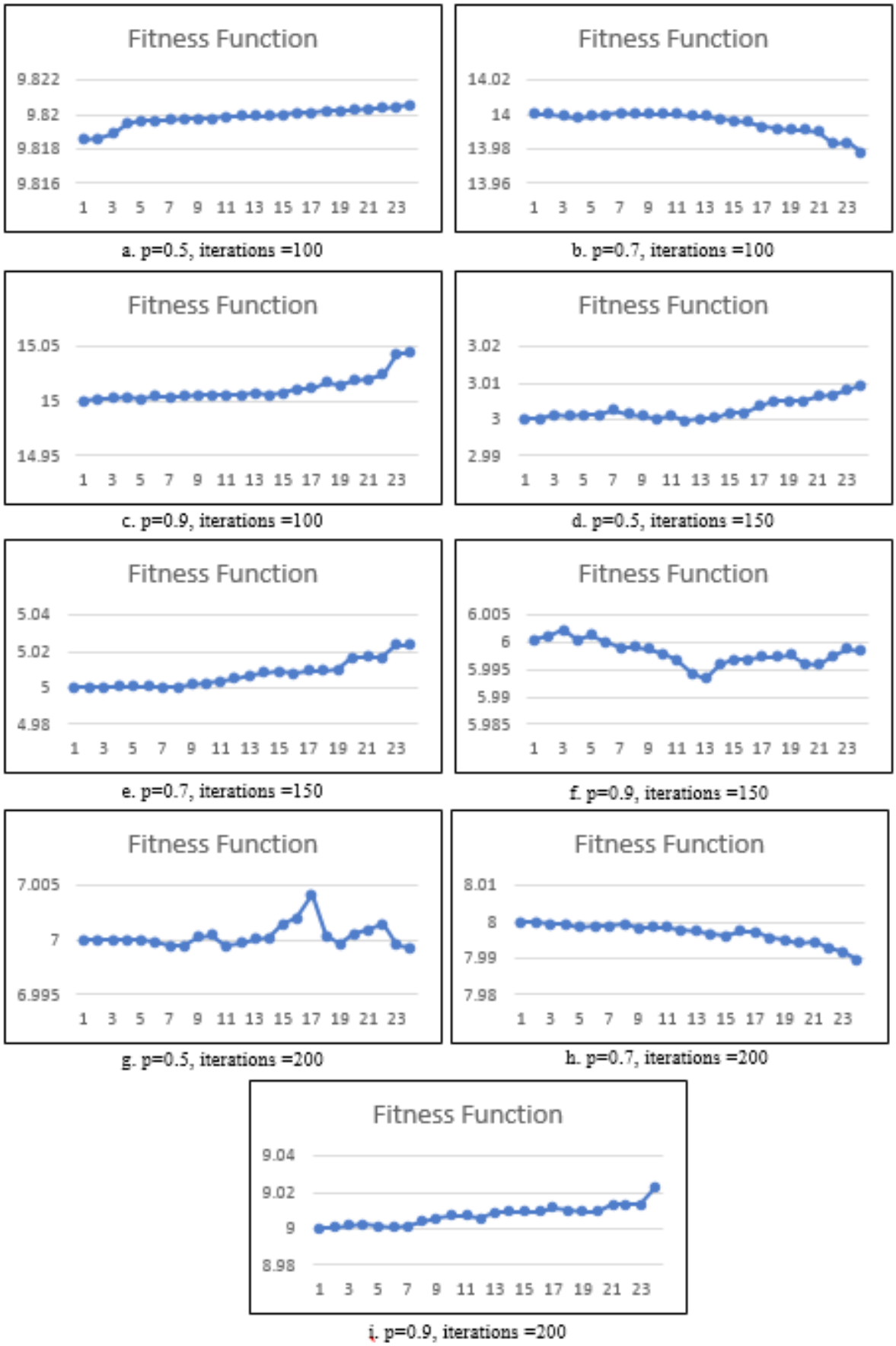

Figure 2: BIA Fitness function of various parameters.

C. BIA Optimization: The resulted images from the former steps are utilized as input to the BIA (these images represent Bat solutions). Several parameters are specified for controlling the keys generation like loudness, pulse rate and frequency, and the number of iterations. With a constant number of iterations, and based on the specified parameters and applying some operations, the evaluation function is utilized in each iteration for generating the key.

D. The Step of Generating Keys: All the numbers that are resulted from the former step are controlled by extracting a unique number. Then, these numbers are multiplied with a specified number for moving the floating-point and getting integer values. The whole numbers are gathered in series and every three digits are separated, after that, the modulation is applied to be considered as values of keys and the carry are concatenated at the sequences end till all the needed keys are generated.

E. Encryption Step: In the step of encryption, the generated key sequence should be equal to the size of information. An operation of exclusive or is applied on the binary representation of these two sequences for obtaining the encrypted text.

The Experiential Results: In the experiential results, several retina images are used for testing and evaluating the resulted sequences of keys. Every color retina image is utilized for producing multi-generation based on several parameters. The results of applying canny edges detection are illustrated in figure 1, where every retina image gives four images that are utilized as bat solutions.

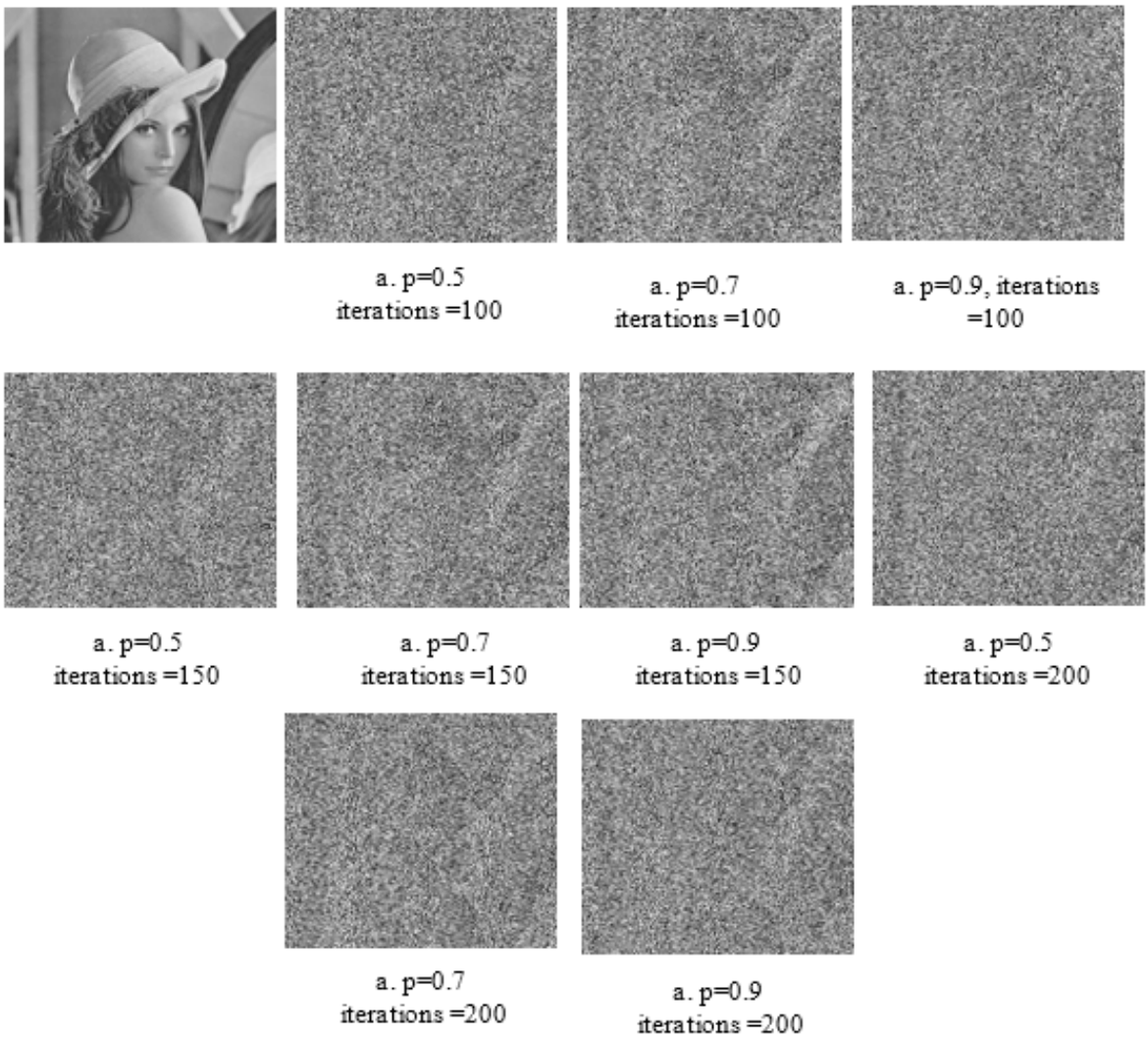

Figure 3: The results of Lena Image Encryption.

For each bat, the fitness function is works on obtaining a unique real number in various behavior even when selecting the same initial parameters. The number of iterations that are selected in this proposed system are; 100,150 and 200. And the selected pulse rates are; 0.5, 0.7 and 0.9, as illustrated in figure 7. The obtained numbers are processed and utilized for generating a key which is utilized later for encrypting the images.

The total size of the key that is needed for encrypting the grayscale image is calculated by multiplying the high and width of the image, i.e. when the size of the image is 256*256, then the total size of the key is equal to the 65536. In order to implement the stream cipher, the input grayscale image is converted into a vector of bits, after that, the exclusive or operation is applied on these two binary sequences (Key, and grayscale image), and the result is reshaped again into a 2D array for representing the cipher grayscale image, as illustrated in figure 3.

In this proposed system, the NIST tests (eleven tests) are utilized for testing the accuracy of nine samples of generated keys. Table 1 shows the results of the tests which demonstrate that the generated keys are accurate and successfully passed all these specific tests.

Table 1: The results of NIST tests.

#	Iteration	Pulse Rates	Block Frequencys	Approximated Entropy	FFTs	Cumulative Sum	Lempel-ZIV Compressions	Frequencys	Non-Periodic Template	Longest Run of Ones	Run Test	Overlap-Template Of All Ones T	Serial Tests
1	100	0.5	0.0280	1.000	0.9542	0.8920	0.78541	0.1116	0.5092	1.000	0.65853	1.000	0.6907
2		0.7	0.5843	1.000	0.4220	0.3997	0.78586	0.4795	0.9992	1.000	0.58028	1.000	0.4985
3		0.9	0.7843	1.000	0.5041	0.7745	0.37889	0.0509	0.988	1.000	0.62032	1.000	0.8422
4	150	0.5	0.1572	1.000	0.6463	0.3145	0.61074	0.1572	1.000	1.000	0.85748	1.000	0.4989
5		0.7	0.5958	1.000	0.4461	0.2995	0.67757	0.3475	0.0449	1.000	0.89413	1.000	0.7605
6		0.9	0.7236	1.000	0.2041	0.0244	0.68037	0.8509	0.0449	1.000	0.42032	1.000	0.8422
7	200	0.5	0.1572	1.000	0.6475	0.2995	0.67889	0.3475	0.9999	1.000	0.89413	1.000	0.7605
8		0.7	0.0215	1.000	0.2041	0.2656	0.68037	0.2341	0.0449	1.000	0.68620	1.000	0.2383
9		0.9	0.5000	1.000	0.6475	0.0291	0.67757	0.0145	0.9987	1.000	0.89413	1.000	0.238321

All the images utilized as a case study in the proposed system are tested for finding their efficiency. Figure 4 shows the histogram analysis test for the original and encrypted images which illustrates there are no relations between the images before and after the encryption. Additionally, the quality measurements (Entropy, SSIM, PSNR, and MSE) are implemented on the resulted encrypted images as shown in table 2. All the obtained results are passed the security analysis and requirement.

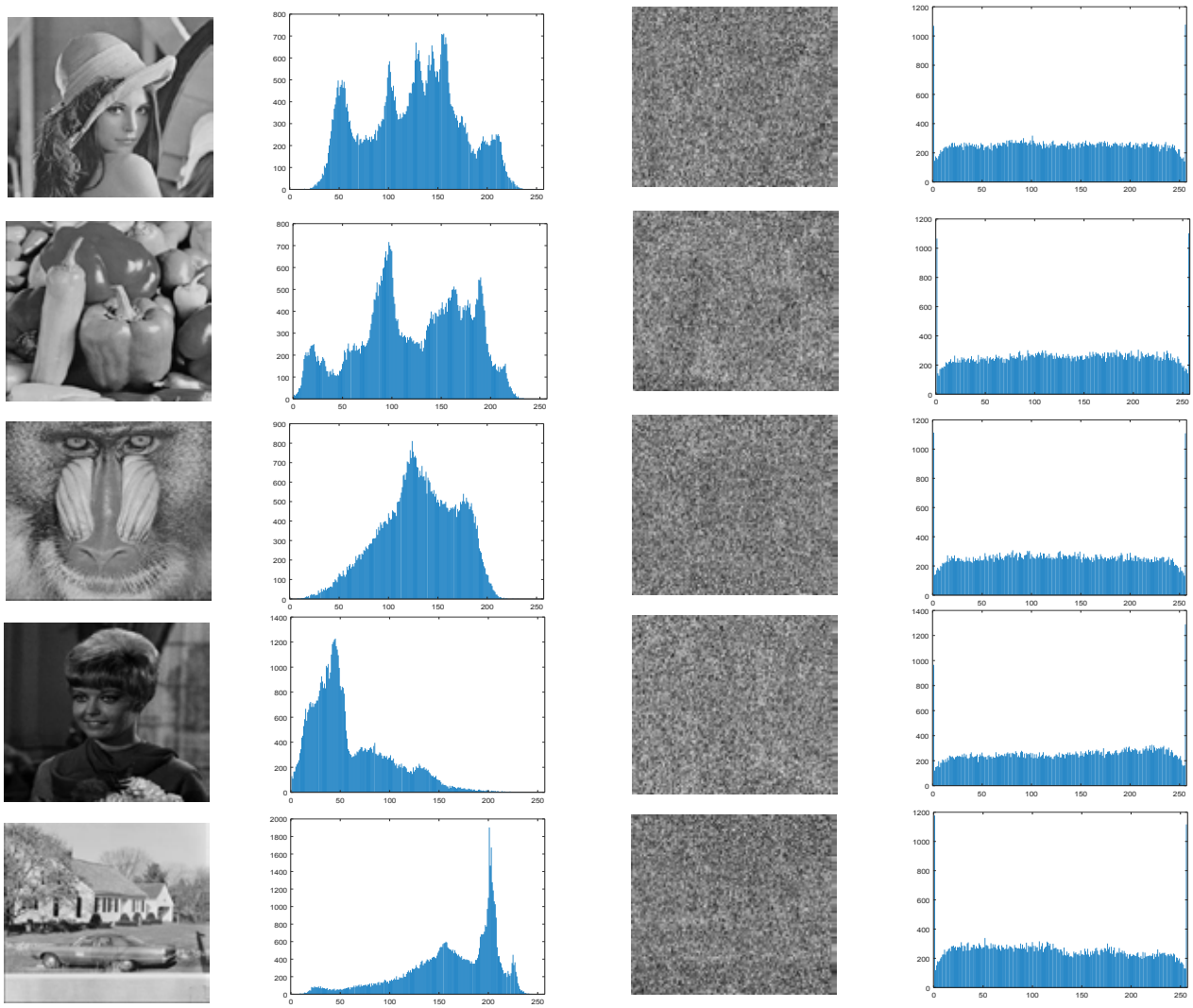

Figure 4: Histogram of original and encrypted images.

Table 2: The main objective measurements of encrypted images.

	Entropy				
	Before	**After**	**SSIM**	**PSNR**	**MSE**
Image#1	7.4683	7.9602	0.0010	9.0035	8179.5533
Image#2	7.6002	7.9570	-0.0060	8.4721	9244.2257
Image#3	7.2615	7.9556	-0.0081	9.5311	7243.8285
Image#4	7.0602	7.9512	0.0041	7.0031	12964.8288
Image#5	7.2820	7.9482	-0.0114	8.1602	9932.5331
Image#6	7.1098	7.9526	-0.0043	7.9811	10350.6837
Image#7	7.2844	7.9590	0.0026	8.8729	8429.2960
Image#8	7.1509	7.9479	0.0015	7.3750	11901.0401
Image#9	7.5004	7.9572	0.0039	9.0532	8086.5365

Conclusion

In this paper, an optimized keys generation system based on the main characteristics (unique features) of the retina image has been proposed. The major idea is to utilize retina images of persons for generating various sequences of keys, and in this way, the significant information concerning each person is kept in secure order. There are lots of steps that are implemented, especially utilizing the canny edge detector on the input retina images. The BIA was utilized as an instance of swarm intelligence to get the behavior of this inspired algorithm in generating keys. The strength of the generated keys is tested using randomness tests and through these tests, all the generated keys were passed successfully. Additionally, the quality measurements (Entropy before and after the encryption, SSIM, PSNR, and MSE) were implemented on the resulted encrypted images regarding the original images, and the obtained results were efficient.

Financial Disclosure: There is no financial disclosure.

Conflict of Interest: None to declare.

Ethical Clearance: All experimental protocols were approved under the College of basic education and all experiments were carried out in accordance with approved guidelines.

References

1. J Wang, J Pan, X Wu. The entropy source of pseudo random number generators: from low entropy to high entropy," 2019 IEEE International Conference on Intelligence and Security Informatics (ISI), Shenzhen, China, 2019: 161-163.
2. AAbusukhon, Z Mohammad, A. Al-Thaher. Efficient and Secure Key Exchange Protocol Based on Elliptic Curve and Security Models," 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), Amman, Jordan, 2019; 73-78.
3. H Boche, RF Schaefer. On the Algorithmic Computability of the Secret Key and Authentication Capacity Under Channel, Storage, and Privacy Leakage Constraints," in IEEE Transactions on Signal Processing. 2019; 67: 4636-4648.
4. Q. Wang, X. Wang, QiuyunLv, Lin You, Wangke Y. Pre-process method for reducing initial bit mismatch rate in secret key generation based on wireless channel characteristics," 2015 IEEE 16th International Conference on Communication Technology (ICCT), Hangzhou. 2015; 888-891.
5. Rizk-Allah RM, Hassanien AE. New binary bat algorithm for solving 0–1 knapsack problem", Complex & Intelligent Systems.2017; 4: 31–53.
6. J Xie, D Wei. Multi-sensor Detection Network based on Improved Bat Algorithm," 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, 2018: 2061-2065.
7. Yang X. A New Metaheuristic Bat-Inspired Algorithm.2010. IEEE, Studies in Computational Intelligence.2010; 65–74 .
8. YechuangW, PenghongW, JiangjiangZ, ZhihuaC. A Novel Bat Algorithm with Multiple Strategies Coupling for Numerical Optimization", Mathematics, 2019; 7: 135.
9. P Topno, G Murmu. An Improved Edge Detection Method based on Median Filter," 2019 Devices for Integrated Circuit (DevIC), Kalyani, India, 2019; 378-381.
10. M Saya. Santhi, "Improved Edge Detection Method in OCT Images using a Hybrid Framework based on CGWO Algorithm," 2019 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 2019; 0465-0469.
11. Panchal P, Bhojani R, Panchal T. An Algorithm for Retinal Feature Extraction Using Hybrid Approach. ELSEVER, Procedia Computer Science.2016; 79; 61–68.
12. JuliánG, Jose M. Hospital Miguel Servet, Mariano RincónZamorano, Margarita Bachiller and EnriqueJ. Carmona Suárez. DRIONS-DB, 2008. http://www . ia. uned.es/~ejcarmona/DRIONS-DB. html