Trustworthy image security via involving binary and chaotic gravitational searching within PRNG selections

Budoor Obid Al-Roithy[†], and Adnan A. Gutub[††]

Umm Al-Qura University, Computer Engineering Department, Makkah, Saudi Arabia

Abstract

In recent years, researchers have been eager to improve image security via various trustworthy encryption models. Normally, image encryption is built on a single PRNG generator which can be considered vulnerable allowing security weaknesses. This research proposed a different image encryption algorithm based on random selection of combining types of efficient PRNGs to further increase the security. Our proposed system focuses on effective keystream generation by using a binary gravitational search algorithm to select random PRNGs starting encryption key followed by chaotic gravitational search algorithm to produce the initial value for the selected generator. The work uses the developed keystream to encrypt images utilizing smart permutation and substitution processes. The experimental results showed acceptability of the proposed method over others. The work evaluation of several statistical measures provided interesting remarks very much pioneering opening interesting directions for future research to come.

Keywords:

image encryptions; PRNG; image scrambling; image shuffling; random number generator.

1. Introduction

Information technology is a group of advanced technologies such as databases, communication networks, and multimedia. The use of these technologies is widespread in the financials, industrial, education, security, and health sectors [1]. Therefore, the security of data against attacks is increasingly important [2]. Information technology needs security technologies such as cryptography, to provide the necessary protection for data from unauthorized access and disclosure [3]. Important information security components are confidentiality, integrity, and availability [2].

Cryptography techniques permit a single person to manage system access and be responsible for the protection of the data inside the system [4]. Data is secured using an information transformation mechanism, known as encryption [5]. The abuse of the encryption key leads to data loss or inaccessibility, which promotes the need for a strong key, to ensure that it is difficult to access protected data [6]. The corruption of encrypted data or key loss results in data loss because the user cannot retrieve the data without utilizing the same encryption key[1].

Image encryption is a process that uses several algorithms to secure an image so it can be transferred over the internet. A significant number of encryption methods have been suggested since the 1970s, some of which have been universally adopted and standardized worldwide [7]. Many of the encryption methods have been designed to secure textual data [8]. There are two types of encryption algorithms: namely symmetric and asymmetric encryption algorithms. Symmetric encryption algorithms involve DES and AES, whereas RSA and ECC are used in asymmetric encryption algorithms [9]. These algorithms are a convenient option for text encryption only, and not appropriate for image encryption because they take too long and are not unsuitable for some of the features of images such as the large capacity of data and redundancy [10].

However, the requirements for image encryption go beyond the capabilities of basic and traditional cryptography algorithms [7]. The application requirements and data structure impose many constraints that should be addressed by the encryption method, for instance, compression efficiency [11], real-time performance [12], format compliance [13], perceptibility [14], complexity [15], and the security level [16]. Progress has been made in building robust security mechanisms for image data to resolve these issues [17] [18].

Using the grid structure and its impact on encryption of digital images as an example, it is clear that the methods for encrypting images differ from those for other data because of different structure. Hence, three different types of operations are used to encrypt images: position permutation, value substitution, and the combination form [7]. Permutation (transposition) is widely found in image encryption methods as primitive operations. This is fundamentally due to the convenience of implementing permutation and its applicability in both the frequency and spatial domains [7]. Furthermore, a highly stable multimedia encryption method can be accomplished by integrating a permutation operation with other basic value substitution operations, such as XOR [7].

In recent years, many different image encryption approaches have been introduced. Random number generators are used to generate encryption keys for use in image encryption and have attracted many researchers interested in improving their performance. Powerful random number generators that fit encryption applications have features and properties that make them robust and unpredictable such as large periods, uniformity, consistency, reproducibility, and independence [19]. So, choosing a weak random generator, e.g. a linear congruential random

Manuscript received December 5, 2020.
Manuscript revised December 20, 2020.
https://doi.org/10.22937/IJCSNS.2020.20.12.18

number generator (LCG), leads to a great chance of prediction and detecting the encryption key [20].

For a PRNG to be used in cryptographic applications, the numbers generated must be cryptographically secure. In most algorithmic PRNGs, the cryptographically secure property is often missing [19]. Because of several vulnerabilities in a generated sequence may easily lead to a critical leak[21], one of our objectives in this paper is to secure the PRNG for producing a reliable random sequence and use PRNG's randomness and unpredictable properties to increase the quality of the PRNG and enhance its statistical properties for image encryption.

Improving the security of image encryption is the primary objective of the proposed technique. To accomplish this purpose, the PRNGs are used with the binary gravitational search algorithm (BGSA) and chaotic gravitational search algorithm (CGSA). In this paper, we suggest a new method for encrypting images by improving defects in generators. Unpredictability is one of the many essential properties and requirements of a PRNG. While a random series may have many other benefits (i.e. good independents, useful length of the period, and uniformity of distribution) it can still be predictable, and hence create risk for secure communication. Therefore, we need to select a generator in an unpredictable way and generate random sequences. We will implement a dynamic system to pick the PRNG as the unpredictable behavior in PRNG for increasing key sensitivity. If we use a single PRNG the hackers may guess the sequences, so instead, we select the PRNG randomly. This means the keyspace and the possible combinations are generated according to the selected seed value and the specific PRNG, but the generation is not limited to a single PRNG. This ensures that the cryptosystem is robust and prevents brute-force attacks by changing the possible key combinations randomly, and making it difficult to determine the key.

The structure of this paper is as follows: related work is presented in section 2; The proposed method is outlined in section 3; section 4 reports and discusses the experimental results; and finally, in section 5, the conclusions are shown.

2. Related Work

One of the simplest approaches to enhance statistical properties is to produce random numbers using combinational chaotic maps. For instance, Wang at el. suggested a PRNG founded on a z-logistic map in 2006 [22]. The new random bit series of a chaotic electronic circuit that is not autonomous was suggested by Ergun and Ozogur in 2007 [23]. Hu et al. suggested a true random number generator (TRNG) through the movement of the PC mouse. Patidar et al created a random bit generator in 2009, founded on two chaotic logistic maps that are generated by assessing the yields of the two chaotic logistic maps [24]. Also, in [25] "A Meta-Level True Random Number Generator" a (TRNG) was proposed by B. Fechner and A. Osterloh.

In recent years, many researchers interested in image encryption have proposed systems that use chaotic systems because of the many advantages and features, i.e. the initial condition is sensitive to changes, randomness, unpredictability, and high complexity [26]. Since 1989, when Matthew proposed [27] the first image encryption system based on the chaotic system, there have been several algorithms published for encrypting images based on chaos theory. For instance, Hua and Zhou [28] proposed an encryption algorithm utilizing a 2-D Logistic-adjusted-Sine map (2D-LASM) as a PRNG; Wang et al. [29] introduced a new PRNG based on a 4-D piecewise logistic map (PLM) with a parameter-coupled piecewise logistic map (PCPLM); then there are four variations of Logistic map as PRNG designed by Ismail et al. [30].

For many years, the reliability of image encryption methods that only use permutation schemes have been examined; most of these schemes have been found to be vulnerable to chosen-plaintext attacks, a consequence of the high redundancy of information in multimedia data and weakness in the encryption algorithms [31] [32].

More recently, several permutation ciphers, including multimedia data protection, have been suggested. Sivakumar and Devi proposed [33] encrypting images with a key by permutation and substitution processes that together serve the aim of the cipher. First, the images are divided into small blocks, then the blocks are mixed and arranged according to the numbers generated by a Lagged Fibonacci Generator (LFG), so the blocks take new locations specified by the random generator's numbers. Furthermore, the same generator is used to generate the XOR key for applying the one-time pad. Two methods for encrypting images were proposed by the authors in [34], and the image in both methods is used as a key. To acquire the encrypted image, the method works on XORed between the pixels of the original image and its key image. In order to ensure images, Kapur, and Paladi [35] suggested a method with two PRNG for encrypting images. They investigated the two methods separately, one method used a Linear feedback shift register (LFSR) algorithm for shuffling the pixels rows and then shuffling the columns of the pixel. The second method used a Blum Blum Shub algorithm to change the intensity of the pixels. Scrambling approaches are used by the writers of [36] to encrypt the gray level; the encrypted image is based on the sequences of the random numbers as a key matrix used to reorder pixel position.

3. The Proposed Method

There are two main methods used to transform an image into its distorted form. The first method scrambles pixel location and the second scrambles pixel value. In the first method, each pixel location of the original image is matched to a new location to obtain the scrambled image. This method does not achieve uniformity distribution because the pixel value is not affected by the encryption method; the pixel value distribution is the same in the original and scrambled images. The second method modifies the value of the pixel in accordance with the scrambling key, and the

value of each pixel of the original image is modified to a new value to obtain the substituted image. The pixel distribution is different in the original and scrambled images. Via an analytical assessment, the second method is more secure than the first, but at the expense of the chance of missing image data. One can merge both methods to successfully obtain a highly secure image encryption algorithm [37]. The implementation phase comprised two methods as follows.

3.1. Encrypt an image by using only a permutation method

The protection of permutation image encryptions has been studied extensively using several steps, as shown in Figure 1.

Step 1. Insert image and determine image size and dimensions.

Step 2. Generate 1-dimension random array with value occurs only once, ranges from one to the original image size. For example, if the user inserts an image with a size of 125 x 125 then the generated array will contain 15,625 values. Save the 1-dimension random array as a secret key, which the two parties (sender and receiver) use to encrypt and decrypt the image.

Step 3. Reshape the original image from a 2-dimensional matrix to obtain a 1-dimensional vector.
Step 4. Mapp each vector element obtained in Step 2 with Step 3, which is the permutation key in Step 2 identical size as the original image will be scrambling with the 1-dimensional original image in Step3.

Step 4. Reshape the obtained cipher image as 2-dimensional array to get the final scrambled image.

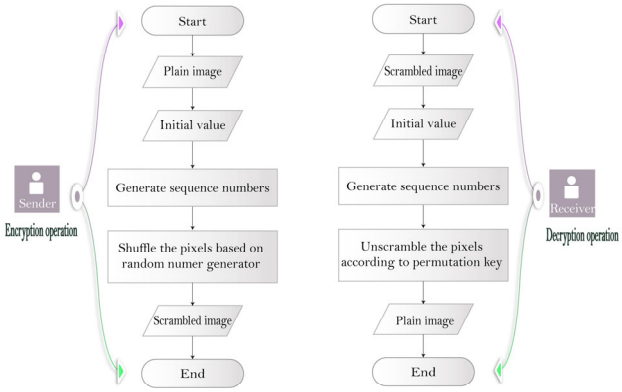

Fig. 1 Encryption and decryption operations of permutation image according to a random number generator

3.2. Encrypt an image using a substitution method

A powerful and secure multimedia cryptosystem can be created by integrating permutation with other basic value transformation processes, i.e. XOR operation. Image entries (or bit-planes) are scrambled in all the popular permutation encryptions by swapping a matrix that is constructed by a PRNG [7]. The XOR operation is described extensively in the following steps, the same used in this study [33]. Figure 2 demonstrates the image encryption steps between the two parties.

Step 1. Insert the scrambled image and determine image size and dimensions.

Step 2. Based on image input and size, generate a sequence of random numbers using a PRNG range from 0 to 255, and utilizing modulus function and its size as the same as the input image size. Save the sequence of random numbers as the XOR key.

Step 3. Perform the XOR operation to create an encrypted image. This operation is applied between the scrambled image and the XOR key.

Step 4. Replace the pixel value array with the intensity value extracted from Step 3.

Step 5. Save the obtained cipher image.

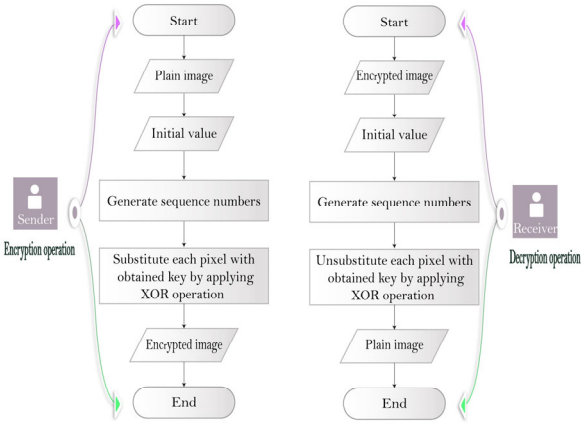

Fig. 2 Demonstration about substitution operation for encrypting and decrypting image.

3.3 The proposed method for improvement

Recently, it has been concluded that using PRNGs alone to encrypt images is not sufficient. Several researchers have used cryptoanalysis to investigate image encryption methods i.e., [38], and [39] were able to entirely reveal and determine the exact permutation mapping. The attack complexity was estimated as O (n · M N) [7]. If the generator used can be predicted, or if it is used as a static default in a cryptosystem, then the security system can be disclosing. These findings strengthen our aim of improving

the encryption system by varying the generator that is used to generate the keys. The improvements made are presented in the next section.

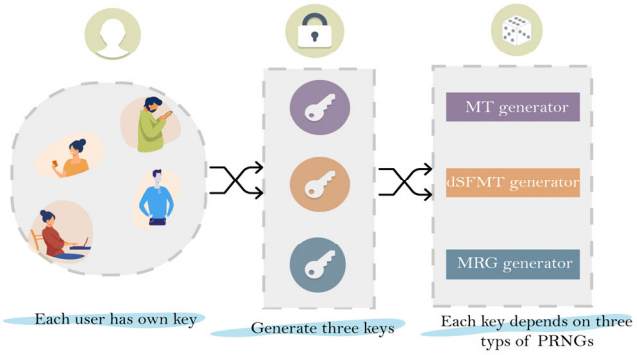

Fig. 3 General idea about distributing generated keys.

The proposed image encryption system is based on the random selection of the generator used to generate the encryption key. The encryption system is unpredictable and non-deterministic. We worked to guarantee that each user will have a different key produced by one of three PRNGs separately, as shown in Figure 3. If there is a brute-force attack on one user, then other users will not be affected because each user has a key from a specific generator.

Therefore, we suggest three efficient PRNGs to generate encryption keys. The three PRNGs types are (1) Mersenne Twister (MT); (2) SIMD-oriented Fast Mersenne Twister (dSFMT); (3) Combined Multiple Recursive (MRG). These generators are randomly selected to generate a key used for encrypting images.

In our suggested method, all generators work independently with their own deterministic equations. Each selected generator produces the key as a single selection for each user. Separation of the three generators ensures that the encryption system works dynamically, and is not depending on only one generator to achieve our objective. For example, K1, K2, and K3 are three keys generated by the MT, dSFMT, and MRG generators, respectively.

The proposed key generation is described in the following steps. The security improvement is divided into two parts: the first part specifies the selection of the PRNG, and the second part considers the initial value.

3.3.1 The first part of the proposed key generation description.

The following steps describe the random selection of the PRNGs.

Step 1. Use a BGSA to select PRNGs randomly. BGSA algorithm produces numbers according to the law of gravity, similar to the approach introduced by Rashedi et al. in [40]. Through this step, the cryptosystem becomes less vulnerable because the keys are not generated by a single generator and there is an increase in complexity. According to the best solution obtained by the BGSA, the system selects one of the three generators to generate an encryption key.

Step 2. Generate K1, K2, and K3 based on the three best solutions of the BGSA. This procedure produces a random generation system. If the best solution equals zero, then K1 is generated using the MT generator. If the best solution equals one, then K2 is generated using the dSFMT generator. Finally, if the best solution equals two, then K3 is generated using the MRG generator.

3.3.2 The second part of the proposed key generation description.

Step 1. Implement a Chaotic Gravitational Search Algorithm (CGSA) [41] to generate the initial value. We select the sinusoidal map based on the GSA algorithm to introduce the initial value for the selected PRNG.

Step 2. Insert the initial value obtained using CGSA into the selected generator. This step is reducing the social engineering if the user inserted a PIN code as the initial value for the PRNGs. Also, it is convenient compared to using a TRNG as an initial value because its speed is very low.

Figure 4 shows the overall processes for generating the encryption key. After we generated a key we use it for encrypting images using permutation and substitution operations. Figure 5 illustrates the whole implementation of the encryption system.

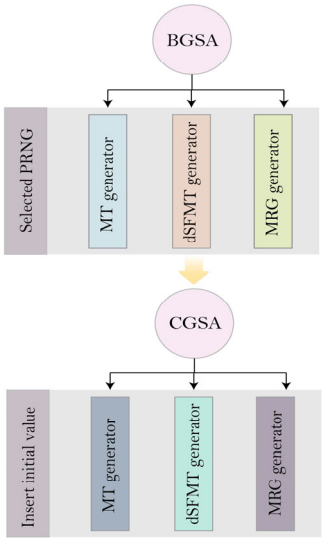

Fig. 4 The procedures of generation encryption key.

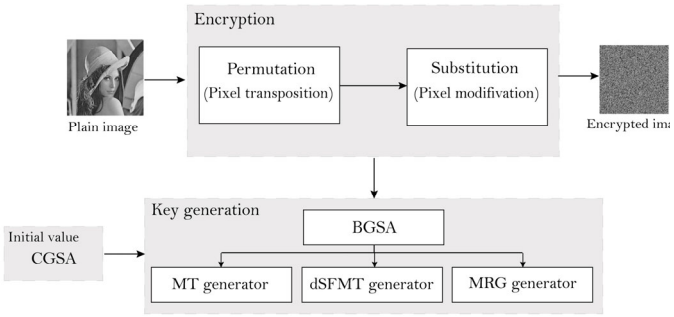

Fig. 5 Block diagram of the proposed encryption system.

4 Experimental Results

The outputs of the encryption system were investigated using several measurements to examine the differences between the original image and its cipher image. The proposed system was implemented using MATLAB R2019a platforms, on operating system Windows 10 Pro, AMD Ryzen 5 3600 6-core processor 3.59 GHz, 16.0 GB memory, and 64-bit operating system, x64-based processer. We use an image database in [42] to apply image quality assessment (IQA) and to extract the good encryption. The selected images from the database were: Lena image with a size 256×256, Baboon image with a size 512×512, and Peppers image with a size 512×512. All the experiments were applied to gray image 8-bit. The security analyses performed were: histogram, entropy, correlation, MSE (Mean-Squared Error), and PSNR (Peak Signal to Noise Ratio).

4.1 Visual examination

All images selected to test the encrypting system are shown in Figure 7. All the encrypted images demonstrated the system's effectiveness at destroying the original image's intensity.

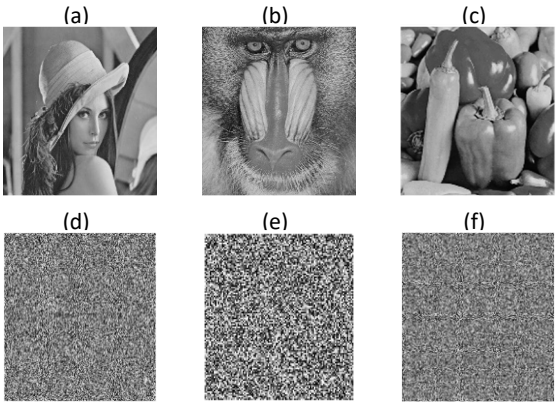

Fig. 6 Display the original images from a to c, and their corresponding is the encrypted images from d to f.

4.2 Histogram

The histogram is a statistical analysis used to assess features of images. It measures the intensity of each pixel according to its rate of occurrence. In order to defend these images against statistical attacks, the information in the image must be spread uniformly. The grayscale histograms of the original and the encrypted image are shown in Figure 8. The difference in the distributions of the original and encrypted images is very clear. The original image histogram shows the actual pixel distribution of the image, whereas the structure regarding the grayscale. Whereas the histogram of the encrypted image shows the uniform distribution of pixels of the image.

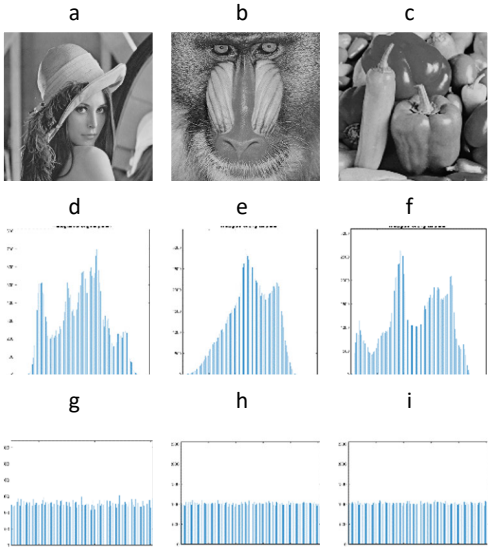

Fig. 7 Histogram of the original and encrypted images. From a to c are presenting the original images horizontally, from d to f are presenting the histogram of the original images, and from g to i are presenting the histogram of encrypted images horizontally.

4.3 MSE (Mean-Squared Error)

MSE is considered to be one of the image error and quality measurements. It computes the difference between original and encrypted images, then computes the power, and after that, the result is divided by the size of the image. Eq.1 explains the computation of MSE [43]. To determine the similarity according to MSE, we should obtain a 0 value. Other than 0 value means that the two images are different. Table 1 shows the experimental results for images encrypted using the proposed system.

$$MSE = \frac{1}{MN}\sum_{X=0}^{M-1}\sum_{Y=0}^{N-1}\left(C(x,y) - P(x,y)\right)^2 \qquad (1)$$

Table 1: MSE results for three gray encrypted images based on three PRNGs selected separately and randomly for generating encryption keys.

Images	Encryption Key	MSE
Lena	MT	7.77e+03
	dSFMT	7.73e+03
	MRG	7.70e+03
Baboon	MT	7.07e+03
	dSFMT	7.03e+03
	MRG	7.03e+03
Peppers	MT	8.45e+03
	dSFMT	8.41e+03
	MRG	8.39e+03

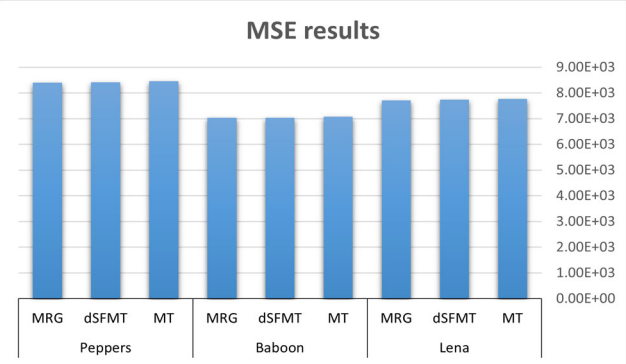

Fig. 8 MSE results for three encrypted images using three types of PRNGs separately.

4.4 PSNR (Peak Signal to Noise Ratio)

PSNR is used to measure the error between two images. A higher PSNR value correlates with high image quality, and a low PSNR value indicates that image quality is destroyed. Eq. 2 was used to estimate the PSNR between the original image and its cipher image.

$$PSNR = 10log_{10}\left(\frac{255^2}{MSE}\right) \quad (2)$$

Table 3: PSNR results for three gray encrypted images based on three PRNGs selected separitly and randomly for generating encryption keys.

Images	Encryption Key	PSNR
Lena	MT	9.2275
	dSFMT	9.2473
	MRG	9.2649
Baboon	MT	9.6361
	dSFMT	9.6416
	MRG	9.6608
Peppers	MT	8.8633
	dSFMT	8.8828
	MRG	8.8908

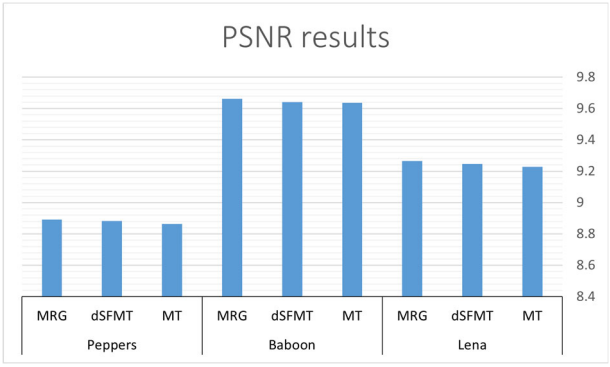

Fig. 9 PSNR results for three encrypted images using three types of PRNGs separately.

4.5 Entropy

Entropy is an analysis of information extracted from part of an image or the entire content. It is a representation of the randomness of the image, and 8 is a satisfactory assessment. Good image encryption should result in a high entropy value. Eq. 3 demonstrates the entropy calculation; p_i indicates the probability of distribution of pixels [44].

$$H(I) = -\sum_{i=1}^{256} p_i \log p_i \quad (3)$$

Table 4: Entropy results for three gray encrypted images based on three PRNGs selected separitly and randomly for generating encryption keys

Images	Encryption Key	Entropy
Lena	MT	7.9972
	dSFMT	7.9969
	MRG	7.9971
Baboon	MT	7.9993
	dSFMT	7.9993
	MRG	7.9993
Peppers	MT	7.9993
	dSFMT	7.9993
	MRG	7.9992

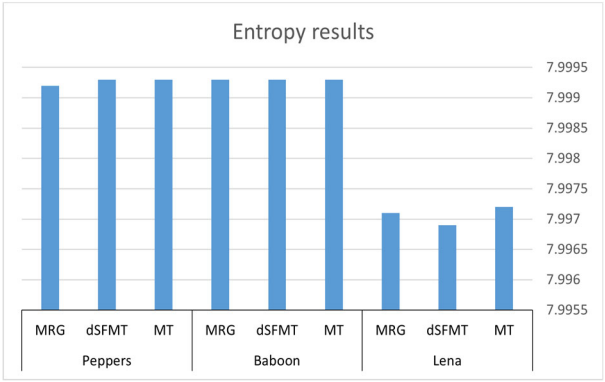

Fig. 10 Entropy results for three encrypted images using three types of PRNGs separately.

4.6 Correlation of Adjacent Pixels

Adjacent pixel correlation is an analysis used to test the strength and weakness of the relationship between pixels in the encrypted image. The strong image encryption system effectively destroys the relationship between adjacent pixels in an image. Values range from 1 to -1 and both ends of the scale indicate a high correlation between adjacent pixels. A high value of 0 indicates a weak correlation. The following equations demonstrate the calculation of correlation among adjacent pixels [45].

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \quad (4)$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N} (x_i - E(x))^2 \quad (5)$$

$$cov(x,y) = \frac{1}{N}\sum_{i=1}^{N} (x_i - E(x))(y_i - E(y)) \quad (6)$$

$$r_{xy} = \frac{cov(x,y)}{\sqrt{D(x).D(y)}} \quad (7)$$

Table 5: Correleation of adjacent pixels results for three gray encrypted images based on three PRNGs selected separitly and randomly for generating encryption keys.

Images	Encryption Key	Orientation		
		Horizontal	Vertical	Diagonal
Lena	MT	-0.0325	-0.0123	-0.0027
	dSFMT	0.0578	-0.0043	-0.0546
	MRG	0.0256	0.0228	0.0816
Baboon	MT	-0.0128	0.0320	-0.0312
	dSFMT	0.0218	0.0174	0.0122
	MRG	-0.0014	-0.0486	-0.0196
Peppers	MT	-0.0300	-0.0113	0.0589
	dSFMT	0.0111	0.0119	0.0310
	MRG	-0.0369	0.0076	0.0023

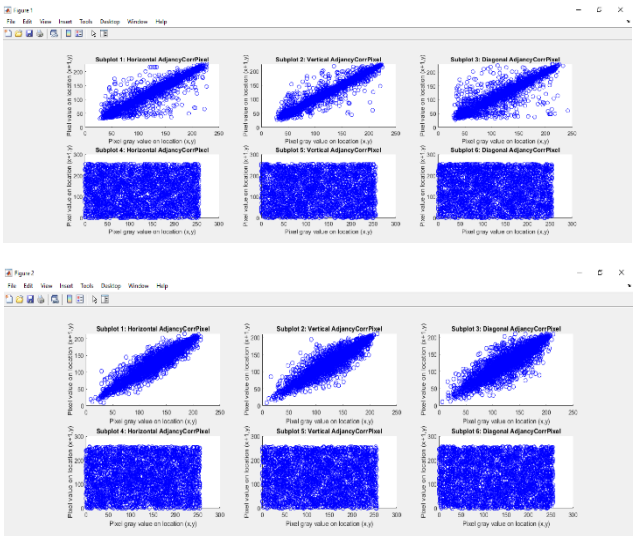

Fig. 12 Three orientations of correlation plots for plain and encrypted Baboon Image.

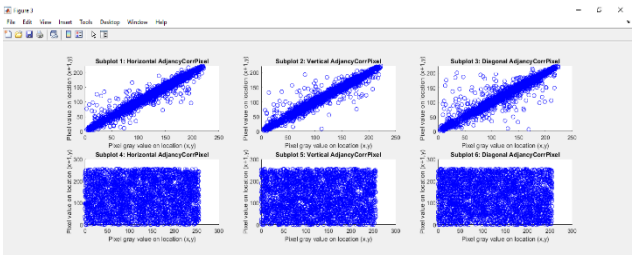

Fig. 13 Three orientations of correlation plots for plain and encrypted Peppers Image.

4.7 NPCR and UACR

The best two measurements utilized to assess the strength and robustness of encryption algorithms are the number of changing pixel rate (NPCR) and the unified averaged changed intensity (UACI). These are used to measure the rate of change that occurs in the plain image. They are powerful techniques because they can detect very small changes, even a change that occurs to a single pixel. A secure encryption algorithm should be close to 100% in NPCE. The following equations describe the calculation of NPCR and UACR.

$$\boldsymbol{NPCR} = \frac{1}{m \times n}\sum_{i=1}^{m}\sum_{j=1}^{n} \delta(i,j) \times 100\% \quad (8)$$

$$\boldsymbol{UACI} = \frac{1}{m \times n}\left(\sum_{i=1}^{m}\sum_{j=1}^{n} \frac{|C_1(i,j) - C_2(i,j)|}{255}\right) \times 100\% \quad (9)$$

Table 6: NPCR and UACR results for three gray encrypted images based on three PRNGs selected separately and randomly for generating encryption keys.

Images	Encryption Key	NPCR	UACR
Lena	MT	99.62	28.65
	dSFMT	99.62	28.54
	MRG	99.57	28.54
Baboon	MT	99.62	27.57
	dSFMT	99.62	27.55
	MRG	99.61	27.48
Peppers	MT	99.60	29.71
	dSFMT	99.61	29.61
	MRG	99.60	29.58

5 Conclusion

This paper introduces a method for encrypting images. The proposed method focuses on the generating of keys using random number generators, which is often a weak point for image encryption. Instead of relying on one random generator, our proposed system uses three separate, effective generators. The benefits of our proposed system are a decrease in the expectation of generating the encryption key, and an increase the complexity. It will be more challenging for a potential attacker to determine which PRNGs are being utilized in our system. Furthermore, we have encrypted images using our system and the experimental results show that our encryption system is effective and accomplished a satisfactory score. In the future, we plan to increase the number of encryption rounds to improve the NPCR and UACI scores.

Acknowledgment

I would like to thank UQU University for giving me an opportunity to be one of their MS students in the college of computer and information systems, and for its efforts in supporting information security fields with expert and inspirational academic staff.

References

[1] A. Al-Qurashi and A. Gutub, "Reliable secret key generation for counting-based secret sharing," *Journal of Computer Science & Computational Mathematics,* vol. 8, pp. 87-101, 2018.

[2] A. Gutub, N. Al-Juaid, and E. Khan, "Counting-based secret sharing technique for multimedia applications," *Multimedia Tools and Applications,* vol. 78, pp. 5591-5619, March 01 2019.

[3] N. Alassaf, A. Gutub, S. A. Parah, and M. Al Ghamdi, "Enhancing speed of SIMON: A light-weight-cryptographic algorithm for IoT applications," *Multimedia Tools and Applications,* vol. 78, pp. 32633-32657, 2019.

[4] K. Alaseri and A. Gutub, "Merging secret sharing within Arabic text steganography for practical retrieval," *International Journal of Research & Development Organisation (IJRDO)-Journal of Computer Science and Engineering,* vol. 4, pp. 1-17, 2018.

[5] M. A. Bani and A. Jantan, "Image encryption using block-based transformation algorithm," *IJCSNS International Journal of Computer Science and Network Security,* vol. 8, pp. 191-197, 2008.

[6] N. A Al-Juaid, A. A Gutub, and E. A Khan, "Enhancing PC data security via combining RSA cryptography and video based steganography," 2018.

[7] A. Jolfaei, X.-W. Wu, and V. Muthukkumarasamy, "On the security of permutation-only image encryption schemes," *IEEE transactions on information forensics and security,* vol. 11, pp. 235-246, 2015.

[8] M. Rani and S. Kumar, "Analysis on different parameters of encryption algorithms for information security," *International Journal of Advanced Research in Computer Science and Software Engineering,* vol. 5, pp. 104-108, 2015.

[9] W. Stallings, "Cryptography and network security principles and practices 4th edition," ed: Pearson Education, Inc, 2006.

[10] S. P. Indrakanti and P. Avadhani, "Permutation based image encryption technique," *International Journal of Computer Applications,* vol. 28, pp. 45-47, 2011.

[11] X. Zhang, Y. Ren, L. Shen, Z. Qian, and G. Feng, "Compressing encrypted images with auxiliary information," *IEEE transactions on multimedia,* vol. 16, pp. 1327-1336, 2014.

[12] H. Cheng and X. Li, "Partial encryption of compressed images and videos," *IEEE Transactions on signal processing,* vol. 48, pp. 2439-2451, 2000.

[13] D. Engel, T. Stütz, and A. Uhl, "A survey on JPEG2000 encryption," *Multimedia systems,* vol. 15, pp. 243-270, 2009.

[14] S. Li, G. Chen, A. Cheung, B. Bhargava, and K.-T. Lo, "On the design of perceptual MPEG-video encryption algorithms," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 17, pp. 214-223, 2007.

[15] D. Engel, E. Pschernig, and A. Uhl, "An analysis of lightweight encryption schemes for fingerprint images," *IEEE Transactions on Information Forensics and Security,* vol. 3, pp. 173-182, 2008.

[16] A. Pande and J. Zambreno, *Embedded multimedia security systems: algorithms and architectures*: Springer Science & Business Media, 2012.

[17] S. Li, G. Chen, and X. Zheng, "Chaos-based encryption for digital image and video," *Multimedia Encryption and Authentication Techniques and Applications,* vol. 129, 2006.

[18] S. Li and R. Lukac, "Perceptual encryption of digital images and videos," in *Perceptual Digital Imaging: Methods and Applications*. vol. 14, ed: CRC Press, 2012, pp. 431-468.

[19] K. Bhattacharjee, K. Maity, and S. Das, "A search for good pseudo-random number generators: Survey and empirical studies," *arXiv preprint arXiv:1811.04035,* 2018.

[20] C. Anley, "Weak Randomness: Part I—Linear Congruential Random Number Generators," *Next Generation Security Software,* 2007.

[21] M. Babaei and M. Farhadi, "Introduction to secure PRNGs," *International Journal of Communications, Network and System Sciences,* vol. 4, p. 616, 2011.

[22] L. Wang, F.-P. Wang, and Z.-J. Wang, "A novel chaos-based pseudo-random number generator," *Acta Physica Sinica,* vol. 55, pp. 3964-3968, 2006.

[23] S. Ergün and S. Özog, "Truly random number generators based on a non-autonomous chaotic oscillator," *AEU-International Journal of Electronics and Communications,* vol. 61, pp. 235-242, 2007.

[24] V. Patidar, K. K. Sud, and N. K. Pareek, "A pseudo random bit generator based on chaotic logistic map and its statistical testing," *Informatica,* vol. 33, 2009.
[25] B. Fechner and A. Osterloh, "A meta-level true random number generator," *International Journal of Critical Computer-Based Systems,* vol. 1, pp. 267-279, 2010.
[26] G. Alvarez and S. Li, "Some basic cryptographic requirements for chaos-based cryptosystems," *International journal of bifurcation and chaos,* vol. 16, pp. 2129-2151, 2006.
[27] R. Matthews, "On the derivation of a "chaotic" encryption algorithm," *Cryptologia,* vol. 13, pp. 29-42, 1989.
[28] Z. Hua and Y. Zhou, "Image encryption using 2D Logistic-adjusted-Sine map," *Information Sciences,* vol. 339, pp. 237-253, 2016.
[29] Y. Wang, Z. Zhang, G. Wang, and D. Liu, "A pseudorandom number generator based on a 4D piecewise logistic map with coupled parameters," *International Journal of Bifurcation and Chaos,* vol. 29, p. 1950124, 2019.
[30] S. M. Ismail, L. A. Said, A. G. Radwan, A. H. Madian, and M. F. Abu-ElYazeed, "A novel image encryption system merging fractional-order edge detection and generalized chaotic maps," *Signal Processing,* vol. 167, p. 107280, 2020.
[31] L. Zhao, A. Adhikari, D. Xiao, and K. Sakurai, "On the security analysis of an image scrambling encryption of pixel bit and its improved scheme based on self-correlation encryption," *Communications in nonlinear science and numerical simulation,* vol. 17, pp. 3303-3327, 2012.
[32] X.-y. Zhao, G. Cheng, D. Zhang, X.-h. Wang, and G.-c. Dong, "Decryption of pure-position permutation algorithms," *Journal of Zhejiang University-SCIENCE A,* vol. 5, pp. 803-809, 2004.
[33] T. Sivakumar and K. G. Devi, "Image Encryption using Block Permutation and XOR Operation," *International Journal of Computer Applications,* vol. 975, p. 8887, 2017.
[34] S. Somaraj and M. A. Hussain, "Securing medical images by image encryption using key image," *International Journal of Computer Applications,* vol. 104, pp. 30-34, 2014.
[35] V. Kapur, S. T. Paladi, and N. Dubbakula, "Two level image encryption using pseudo random number generators," *International Journal of Computer Applications,* vol. 115, 2015.
[36] M. M. Aziz and D. R. Ahmed, "Simple image scrambling algorithm based on random numbers generation," *Int J,* vol. 5, 2015.
[37] R. Shelke and S. Metkar, "Image scrambling methods for digital image encryption," in *2016 International Conference on Signal and Information Processing (IConSIP)*, 2016, pp. 1-6.
[38] S. Li, C. Li, G. Chen, N. G. Bourbakis, and K.-T. Lo, "A general quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks," *Signal Processing: Image Communication,* vol. 23, pp. 212-223, 2008.
[39] C. Li and K.-T. Lo, "Optimal quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks," *Signal processing,* vol. 91, pp. 949-954, 2011.
[40] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "BGSA: binary gravitational search algorithm," *Natural Computing,* vol. 9, pp. 727-745, 2010.
[41] S. Mirjalili and A. H. Gandomi, "Chaotic gravitational constants for the gravitational search algorithm," *Applied soft computing,* vol. 53, pp. 407-419, 2017.
[42] *Image Databases*. Available: http://www.imageprocessingplace.com
[43] "An Enhanced Least Significant Bit Steganography Technique," 2016.
[44] Y. Wu, J. P. Noonan, and S. Agaian, "Shannon entropy based randomness measurement and test for image encryption," *arXiv preprint arXiv:1103.5520,* 2011.
[45] Y. Li, C. Wang, and H. Chen, "A hyper-chaos-based image encryption algorithm using pixel-level permutation and bit-level permutation," *Optics and Lasers in Engineering,* vol. 90, pp. 238-246, 2017.

Author biography

Budoor Obid AlRoithy is currently a graduate student, pursuing Master of Sciences (MS) degree in Computer Sciences & Engineering from Umm Al Qura University (UQU). Her MS program at UQU is specialized in the information security path offered by the College of Computer and Information Systems offered at UQU-Makkah Campus, Saudi Arabia. Her field in MS thesis revolves around Cryptography. She received her BSc in computer science from the Department of computer science and engineering, Taibah University, Madinah, Saudi Arabia. She worked in information technology (IT), and Biostatistics Departments of the Madinah Health Affairs Directorate. She worked as a penetration tester for the information technology Department of the Ministry of Hajj and Umrah at al-Madinah al-Munawwara, Saudi Arabia.

Adnan Abdul-Aziz Gutub is ranked as Full Professor in Computer Engineering specialized in Information and Computer Security within College of Computers and Information Systems at Umm Al-Qura University (UQU). He has been working as the general supervisor of UQU scientific council following his assignment as Vice Dean of the Custodian of the Two Holy Mosques Institute of the Hajj & Omrah Research, Known publicly as Hajj Research Institute (HRI), within (UQU), Makkah Al-Mukarramah, all Muslims religious Holy City located within the Kingdom of Saudi Arabia.

Adnan's academic experience in Computer Engineering was gained from his previous long-time work as Associate Professor, Assistant Professor, Lecturer, and Graduate Assistant, all in Computer Engineering at King Fahd University of Petroleum and Minerals (KFUPM) in Dhahran, Saudi Arabia. He received his Ph.D. degree (2002) in Electrical and Computer Engineering from Oregon State University, USA. He had his BS in Electrical Engineering and MS in Computer Engineering both from KFUPM, Saudi Arabia.

Adnan's research work can be observed through his 115+ publications (international journals and conferences) as well as his 5 US patents registered officially by USPTO. His main research interests involved optimizing, modeling, simulating, and synthesizing VLSI hardware for crypto and security computer arithmetic operations. He worked on designing efficient integrated circuits for the Montgomery inverse computation in different finite fields. He has some work in modeling architectures for RSA and elliptic curve crypto operations. His interest in computer security also involved steganography and secret sharing focusing on image-based steganography and Arabic text steganography as well as counting based secret sharing.

Administratively, Adnan Gutub filled many executive and managerial academic positions at KFUPM as well as UQU. At KFUPM - Dhahran, he had the experience of chairing the Computer Engineering department (COE) for five years until moving to Makkah in 2010. Then, at UQU - Makkah, Adnan Chaired the Information Systems Department at the College of Computers & Information Systems followed by his leadership of the Center of Research Excellence in Hajj and Omrah (HajjCoRE) serving as HajjCoRE director for around 3-years until the end of 2013. Then, he was assigned his last position (until March 2016) as the Vice Dean of HRI, i.e. the Custodian of the Two Holy Mosques Institute of the Hajj & Omrah Research.