

# Varying PRNG to improve image cryptography implementation

Adnan Gutub\* and Budoor Obid Al-Roithy

*Computer Engineering Department, College of Computer & Information Systems, Umm Al-Qura University, Makkah, Saudi Arabia*

*\*Corresponding Author: aagutub@uqu.edu.sa*

**Submitted:** 01/04/2020

**Revised:** 16/11/2020

**Accepted:** 19/12/2020

## ABSTRACT

Securing information became essential to exchange multimedia information safely. These exchanged data need to be transformed in a well-managed, secure, and reliable manner. This paper focuses on securing multimedia images via cryptography during transmission among users using an effective selection from several Pseudorandom Number Generators (PRNG). This paper implements several PRNG techniques involved within consecutive cryptoprocesses of substitution and transposition that have proven a secure process. In the study, different PRNGs are tested to encrypt images in forms of grayscale and colored RGB images compared to current similar approaches. The work experimentation is aiming at investigating and identifying suitability and reliability through security measures standard parameters. The research is showing proper PRNG selection as an attractive, significant work worth remarking for image cryptography.

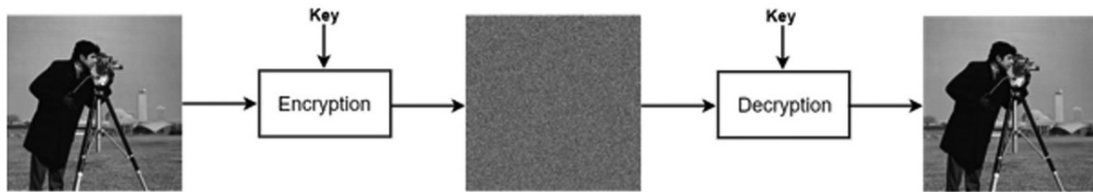
**Keywords:** Digital image; Image encryption; Image scrambling; Image shuffling; Pseudorandom number generator (PRNG).

## INTRODUCTION

Everyone is turning to the digital world, which considers the Internet an essential part of communication between all parties. Internet communications are vulnerable to violations in the security triangle, which needs proper ensuring information security (Almutairi et al., 2020). Any data transfers through normal insecure communication channels are an issue to an attack affecting Confidentiality, Integrity, and Availability (CIA), which are the main threats considered in information security. Securing these CIA three components of information security has become an urgent need (Gutub et al., 2019).

Today, our living in digital world puts all communication multimedia to be facing urgent security challenges (Gutub and Fattani, 2007). The digital media includes several types, such as audio, video, photo or images, holograms, social media, and any virtual reality multimedia (Kheshafaty and Gutub, 2020). It is found in many industries associated with content of education (Almutairi et al., 2020), health (Bin-Hureib and Gutub, 2020), and government sharing services (Gutub and Al-Qurashi, 2020). This leads to confirm the digital community needs for a continues secure environment that guarantees to deliver data for the user and exchanges in a safe manner (Alharthi and Gutub, 2017). In fact, the usage of images has become more widespread among users nowadays. So, encryption data are necessary to protect the privacy of content and prevent unauthorized control or access information by modifications (Hassan and Gutub, 2020). This implies the necessity of checking the image's authentication via passwords to guard them as vital multimedia data (Al-Juaid et al., 2018).

Currently, users' sensitive images are protected via cryptography (Alharthi and Gutub, 2017). The use of encryption for that data is urgently needed to be protected when transmitted through unsafe Internet channels (Gutub, 2011). Cryptography converts data to another form such that it changes texts from plain texts to ciphertexts form (Al-Otaibi and Gutub, 2014). Encryption is making many different mathematical calculations to substitute the data and convert it into useless information (Gutub and Tenca, 2004). It needs a secret key for encrypting and decrypting the data to ensure security (Al-Juaid and Gutub, 2019). Figure 1 describes the standard overview analogy of image encryption; it shows a plain image converted to cipher image using the same symmetric key, known as symmetric (secret) key image cryptography.



**Figure 1.** Procedures for encrypting and decrypting images.

Most applications involve digital images in secure scenarios. It has recently entered heavily into sensitive fields like legal, military, and medical systems (Sharma and Kowar, 2010). The crypto methods designed for encrypting images were limited and not very suitable for today's usages (El-Samie et al., 2013). For example, the AES system as is not an effective, efficient scheme to encrypt images. The structure of the images is different from AES practical data patterns. For this reason, researchers study the AES cipher scheme's primary operations to be tailored and organized for encrypting images (Saha et al., 2018). This caused the focus of image cryptography to benefit from chaos theory and permutations and substitutions as an effective way to be applied for images security (El-Samie et al., 2013).

In this paper, we introduced exploring different pseudorandom number generator (PRNG) options for possible improving the image encryption security strategy (Saha et al., 2018). Our research tested its experimentations on grayscale and RGB images due to their popularity. The work explored the image encryption involving the different PRNG within both permutation and substitution sequential techniques, for fair comparisons, i.e., similar in principle to the novel two steps security presented in (Saha et al., 2018), as shown in the illustration in Figure 2. To be specific, the research's main contribution implemented the crypto approaches via creating the key streams by six pseudorandom number generator (PRNG) for selecting the appropriate one. The six PRNG models involved are: (1) Mersenne Twister, (2) SIMD-oriented Fast Mersenne Twister, (3) Combined Multiple Recursive, (4) Multiplicative Lagged Fibonacci, (5) Legacy MATLAB 5.0 normal generator algorithm, and (6) Modified subtract with borrow generator. According to this sequence, the pixel permutation and substitution are performed differently, i.e., six times applying every PRNG showing different results. Applying permutation and substitutions aims to increase the complexity by combining the two techniques, reduce the possible correlations between the plain images and the cipher images, stand powerful against common differential attack measures, and prevent brute force attacks. The work testing steps on the two image types are shown in Figure 3.

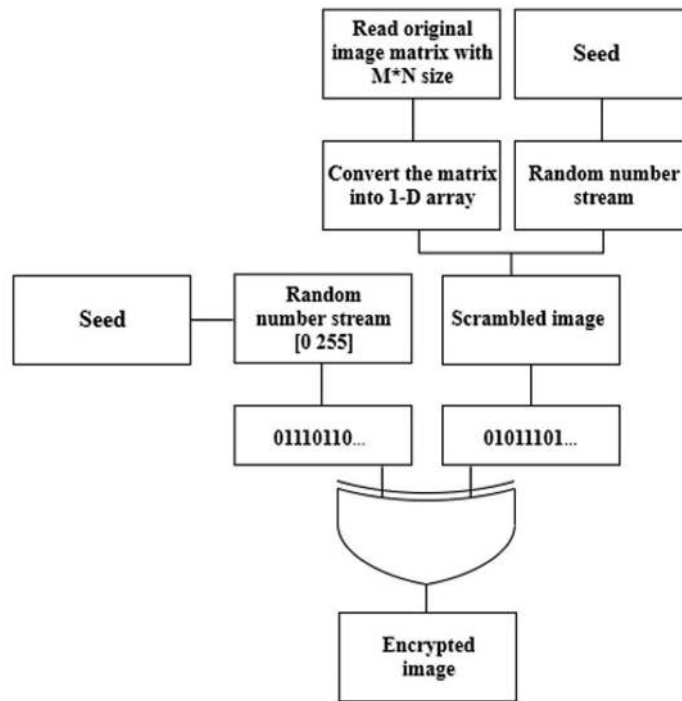


Figure 2. The path of encrypting images by using PRNG sequentially.

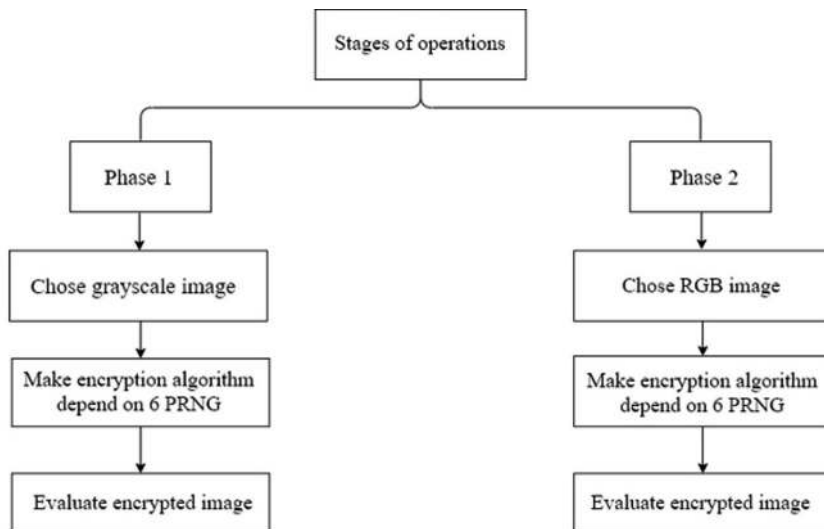


Figure 3. Procedure steps of testing operations.

The flow of the paper presentation is as follows. Section 2 covers the literature review. After that, Section 3 describes the methodology of the proposed image encryption mechanism using the PRNG algorithm. Then, Section 4 discusses the results and shows the experimental results. Next, Section 5 presents the comparisons, followed by Section 6, that concludes the paper.

## LITERATURE REVIEW

The researchers in the digital security field have proposed many digital encryption methodologies that specialize in media encryption and decryption mechanisms. This section gives a general overview of media encryption methodologies, where PRNG algorithms are used to encrypt the digital images.

In 2018, a novel PRNG based image encryption methodology was proposed by Saha et al. (2018). The proposed encryption methodology showed highly efficient encryption processing by incorporating the two processes permutation of pixels positions and substitution of pixels positions. Therefore, the proposed encryption methodology is considered a two-operation image encryption, where in the first operation the shuffling of the pixels is based on the PRNG algorithm LFSR. After that comes a second operation, where the XOR is applied on the pixels to substitute their values by replicating the rows with columns, i.e., to produce the final encrypted image.

In 2017, a new image encryption methodology was proposed by Sarma and Lavanya (2017) based on scrambling techniques to change pixels' positions. The methodology is based on two encryption keys used as input to their proposed algorithm to produce a sequence of random numbers that will be used to scramble the image to be encrypted. Their methodology reads the image, finds its size, converts it from a 2D matrix to a 1D array, generates the random sequence of numbers based on the given two secret keys, scrambles the positions of pixels accordingly, and outputs the encrypted image. For decryption, the previous steps are applied in reversed order to obtain the plain-image again at the receiver end. Three statistical metrics are tested for the images before encryption, after encryption, and after decryption to evaluate image encryption experimentation. The metrics are the MSE, the PSNR and SSIM, and the results show that the encryption methodology is working correctly and efficiently.

In 2015, Ramesh and Jain (2015) proposed a new hybrid image encryption methodology using Pseudorandom number generators. Their methodology combined two pseudorandom number generator algorithms; the first is the Altered Sophie Germain Prime (ASGP), where the generated pseudorandom numbers are used as the new values for each pixel of the image to be encrypted to output an intermediary ciphered image. After that comes the second phase, where Lehmer Random Number Generator (LRNG) is used to generate pseudo random numbers based on the user's entered key. The resulting sequence of random numbers is used to swap each pixel's position of the encrypted image with another pixel. They are hence create a two-operation encryption process to increase the difficulty of image cryptanalysis. For experimentation analysis, they have used the visual test, the histogram test, the entropy, the key sensitivity, and the coefficient of correlation. The results showed that their proposed algorithm has linear complexity, making it fast yet easy to implement.

Another novel algorithm was proposed in 2015 by Kapur et al. (2015) to encrypt and secure images using other PRNG. Their proposed methodology is a two-operation image encryption algorithm. The first phase uses the Linear Feedback Shift Register algorithm to swap the rows and then swap the image columns to encrypt it to produce an intermediary ciphered image. After that comes the second phase, where the Blum Blum Shub algorithm is used to substitute the values of the intensity of each pixel of the image using the generated pseudorandom numbers and producing the final ciphered image. To test the quality and efficiency of their implementation and to analyses the resulting ciphered images, they have used the three tests of the correlation coefficient, the entropy and the Peak Signal to Noise Ratio (PSNR), in which they have given results indicating that the resulting ciphered images are hard against cryptanalysis attacks.

In Rohith et al. (2014), the authors suggested a composite method to generate the encryption key used to encrypt the gray image. The method depends on the image encryption using the theory of chaos. Through two random generators, Logistic map and LFSR, they are used to generate two keys and apply the XOR process to obtain the final key that shall be used to encrypt the image.

In Banthia and Tiwari (2013), the study shows the encryption of images using random generators. They identified two random generators to encrypt images, linear congruential generator, and Logistic map. All methods follow the traditional methods of cryptography, namely transposition and substitution, to shuffling rows, columns, and masks between rows and columns. Moreover, apply image quality measures.

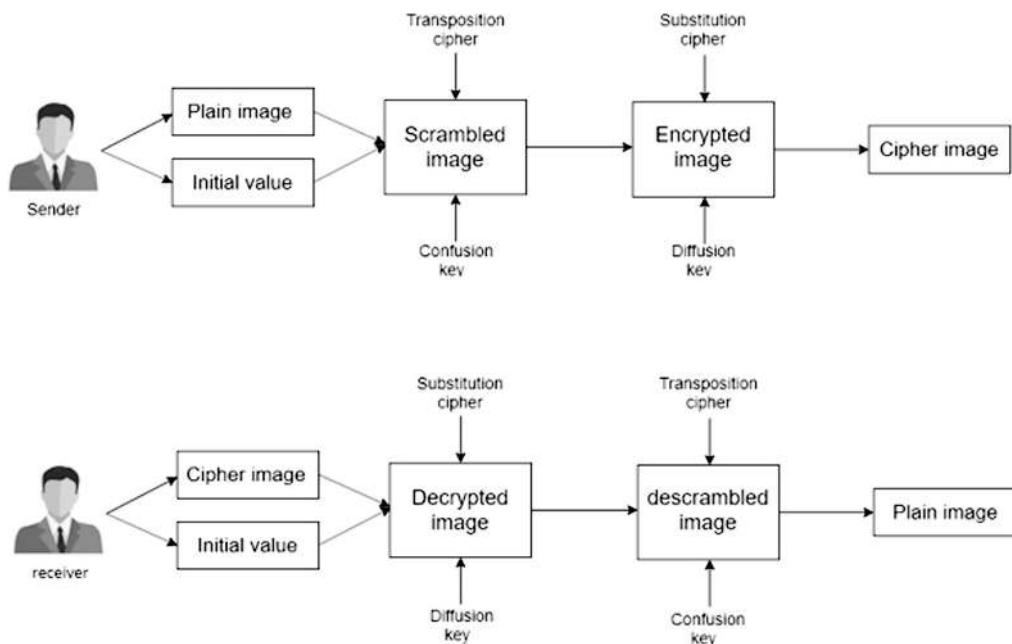
## PROPOSED METHOD

This paper proposes different keys PRNG based image encryption that encrypts two samples of digital images RGB and grayscale images. The encryption process makes at the sender side, and the decryption process at the receiver side.

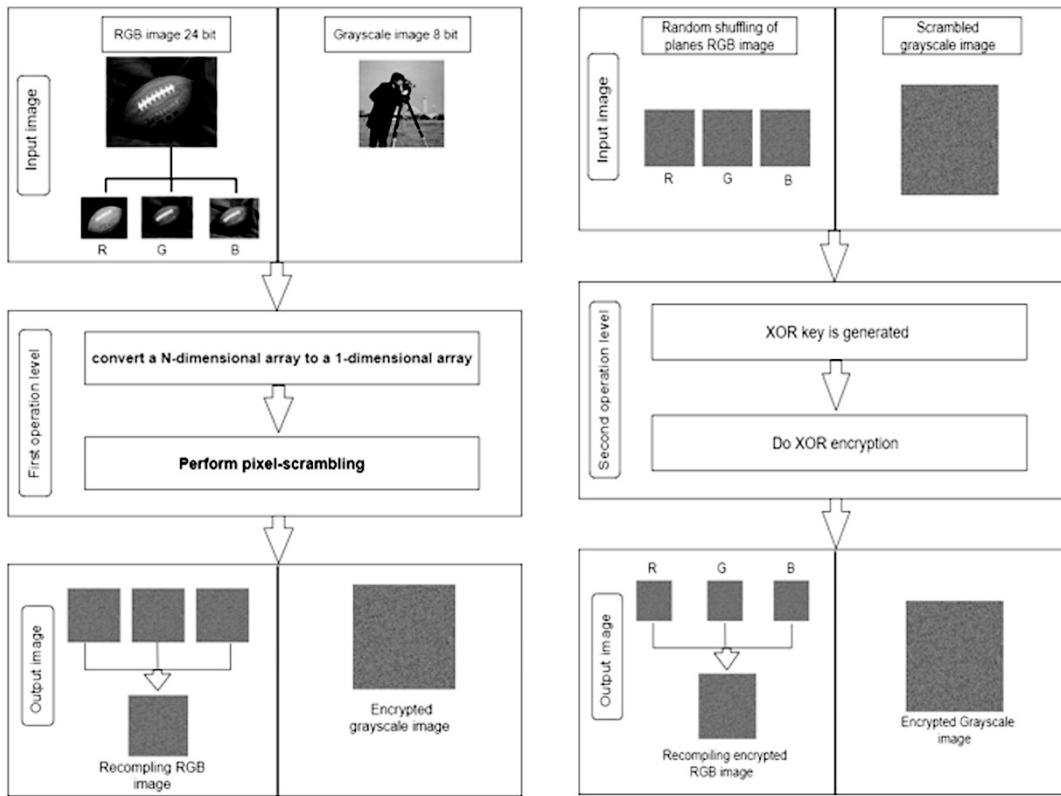
At the encryption process, the proposed methodology receives the image to be encrypted, and the secret pin-code is used as a secret password only between the authorized users (the sender and the receiver). The proposed methodology takes the pin-code as the seed value to the PRNG algorithm to encrypt the image and outputs the encrypted image.

On the other hand, the proposed methodology works in the reversed order at the receiver side, where it receives the same pin-code from the receiver user and the encrypted image, takes the pin-code from the receiver user, and inserts it as the seed value to the PRNG. However, these time steps taken previously to encrypt the image are now taking in the reversed order to decrypt the image and output the decrypted image to the receiver user.

Figure 4 describes the steps of sharing images in a general way between the two ends of the communication while transferring images over the network. The sequence of the operations at the sender side begins with the transposition process and is followed by the substitution process, while the sequence of operations on the receiver side is in reverse order.



**Figure 4.** General steps encryption and decryption processes.



**Figure 5.** Transposition and substitution encryption.

Figure 5 illustrates the image encrypting operations. The left blocks illustrate the transposition operation, the first block describes the system when it reads the 24-bit color and 8-bit gray images, whereas the second block explains the conversion image from two or three dimensional to one dimensional. After that, the one-dimensional array is mapping according to the vector of a sequence of random numbers that we obtained from the PRNG and used as the permutation key to reposition the pixels. The final block explains the results we got from the transposition cipher that appeared as a scrambled image. The right blocks illustrate the substitution operation, the first block describes the system when it reads the scrambled image and it is either 24-bit color or 8-bit gray images, which we got from the previous operation. The second block demonstrates the usage of the generated XOR key for applying the XOR operation to the scrambled image. The final right block shows the final results of the ciphered image.

The following subsections explain the steps to be considered during image encryption and decryption.

### Encryption Process

In the encryption process, we study the two operations performed within the cryptosystem. The process runs transposition, followed by substitution, which has been proven to secure the images for communications (Saha et al., 2018). The cryptography algorithm is tested on single plane grayscale images as well as RGB images, an analogy shown in Figure 5.

#### First operation: transposition

**Step 1:** Read the plain image from the user, a grayscale image or an RGB image.

**Step 2:** Get the Secret pin (Personal Identification Number) code from the user as the seed of PRNG.

**Step3:** Divide the image into separate blocks, where the number of pixels in each block is one pixel per block.

**Step 4:** Test the six PRNG algorithms, starting them with the same pin code (seed) to generate a random sequence. The six PRNGs considered in this study: (1) Mersenne Twister (2) SIMD-oriented Fast Mersenne Twister (3) Combined Multiple Recursive (4) Multiplicative Lagged Fibonacci (5) Legacy MATLAB 5.0 normal generator algorithms, and (6) Modified subtract with borrow generator.

**Step 5:** Shuffle the pixel's positions according to the PRNG results (generated random numbers). The pixel permutation is performed six times by the separated PRNG.

**Step 6:** Display the scrambled image after transposition and examine the security of the results.

### **Second operation: substitution**

**Step 1:** Use the same random keys of PRNG for every transposition results. The range of each key from the six PRNG is (0 to 255). The key size is being the same as the image for applying the one-time-pad XOR substitution operation.

**Step 2:** Do XOR the resultant image with a key to encrypt the image. An XOR operation is substituted with each pixel value in the image with the pixel value's corresponding key value.

**Step 3:** Display the encrypted image after substitution and examine the results.

### **Decryption Process**

The decryption process applies the encryption algorithm in reverse order. The process takes place in the communication process at the receiver end of the image transmission. The original image can be obtained by entering the user with the correct pin code used for the same image encryption. To be specific, the decryption process can be outlined as below.

#### **First decryption operation: XOR (de-substitution)**

**Step 1:** Read the encrypted image from user and get the length, breadth, and num of channels.

**Step2:** Generate the same sequence number key by a similar process that was used for encryption.

**Step 3:** Perform XOR decryption between the image and its key-image.

#### **Second decryption operation: pixel unscrambling (de-transposition)**

**Step 1:** Use the same pin code with the same PRNG algorithms to generate the six PRNG same sequences of random numbers.

**Step 2:** Use the generated random numbers in the reversed order to un-scramble the pixels of the encrypted image (de-transposition: undo the shuffling of pixels positions).

**Step 3:** Convert the sets of unscrambled pixel blocks to an image. Then, retrieve and display the decrypted image.

The following algorithm is a pseudocode for encrypting image by using MT generator. It describes the combination of two encryption operations.

---

 Algo (1) Encrypt image by using MT generator
 

---

ALGORITHM: Permute block and Substitute pixel Image Algorithm.

Input: PIN cod, RGB image, number of blocks (Noblock).

Output: Ciphared image

1. Ensert two inputs: image size, and seed value.
  2. Generate x sequence by using MT generator.
  - Divide image into many blocks%
  3.  $i = 0$  counter row image.
  4. For  $r = 1$ : Noblock: length–Noblock+1
  5. Increment i row.
  6. Start  $j = 0$  counter column image.
  7. For  $c = 1$ : Noblock:width–Noblock+1
  8. Increment j column.
  9. Stor each block in a different new variable b.
  10. End for.
  11. End for.
  - % Permutation operation (Reshaping matrices by converting 2D into 1D )
  12. For  $n = 1$  to length b.
  13. Swap blocks of image with index x sequence.
  14. End for.
  15. Convert cell2mat and save image as scrambled image.
  - % Substitution operation
  15. Generate y sequence with the interval [0 255] by using MT generator.
  16. For  $L = 1$  to length direction of scrambled image.
  17. For  $W = 1$  to width direction of scrambled image.
  18. [xored image] [Scrambled image] XOR[y].
  19. End for.
  20. End for.
- 

## RESULTS AND DISCUSSION

The section studies the model's results and statistical comparison remarks applying the approach images cryptography. The testing is noted at every stage of encryption to build a fair exploration. The study tested its philosophy on the two forms of images, i.e., grayscale images and color RGB images. Our security analysis is made by studying several metrics, showing results of key sensitivity, visual testing, histogram, entropy, correlation, MSE (Mean-Squared Error), PSNR (Peak Signal to Noise Ratio), structural similarity index (SSIM), mutual information (MI), and measuring time analysis.


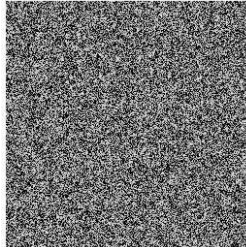

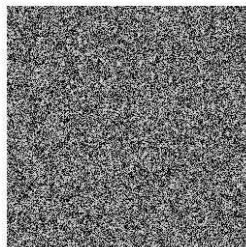


The implementation and examination platform used was MATLAB version number R2019a, on a laptop with an operating system of Windows 10, Intel 5(R) Core(TM) i5-7200U running CPU @ 2.50GHz 2.70GHz, 8 GB memory, and 64-bit operating system, x64-based processor, in order to implement the methodology, and conduct the images encryption and decryption, as well as doing all the testing that comes after finishing the implementation phases. Experiment sampling results have been generated via applying the encryption proposal using the same size methodology on the two different formats of grayscale and RGB images of size 256×256 as TIF format, giving remarkable metric results as follows.

### Key Sensitivity

The entered pin code will be used as the seed value for the PRNG algorithm, and during implementation, this pin code will produce a sequence of random numbers based on this seed input value. The length of the seed should be less than  $2^{32}$ , in other words, the user can enter a pin code with a length up to 10 integers; usually, four integer numbers are the standard length of use. The seed's main advantage is that the produced sequence of numbers produced to encrypt the image can be reproduced later by the receiving user to decrypt and receive the encrypted image. In addition, each time the user enters a different pin code, a different sequence of random numbers will be produced accordingly. Hence, the output of the encrypted image will be different and unique from any previous encryption processes. This means that if the user entered the pin code incorrectly, he/she could not read the image and decrypt it. Only the user who knows the secret pin code you used to generate the image will could decrypt it. When the correct pin code is provided, then the same shuffled and substituted sequence will later be obtained. The following Table 1 confirms when the user decrypted image with the wrong key.


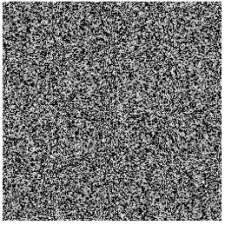

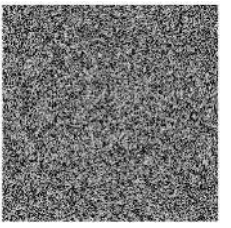
**Table 1.** Key sensitivity.

Original image	Encrypted image
	
Decrypted image with correct key	Decrypted image with wrong key
	

### Visual Testing

As shown below in the following images, by the visual testing with abstract sight, nothing can be inferred from the encrypted images, proving that the image encryption methodology is visually compelling. Table 2 shows sample images, in which we validated our algorithm and presented each original image with corresponding encrypted images of the operation's steps. The study considered many images to prove its experimentations and phenomena but presented a simple image's detail to simplify the reading flow and understanding representation.

**Table 2.** Gray and RGB images of visual testing.

Gray image samples		Color image samples	
Original image	Encrypted image	Original image	Encrypted image
			

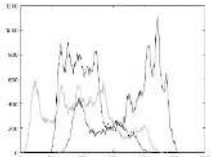
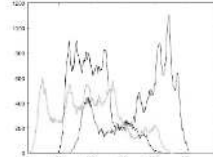
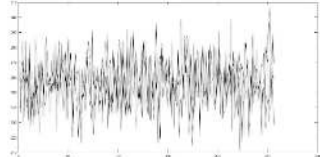
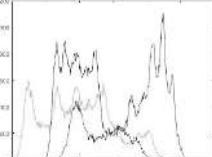
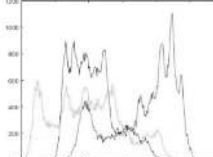

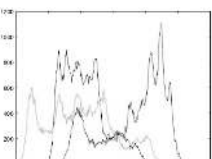
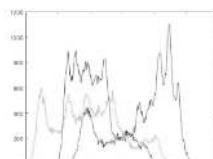
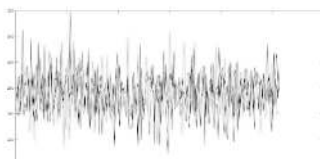
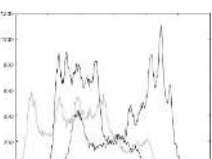
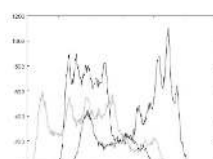
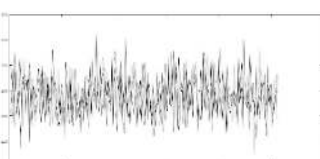
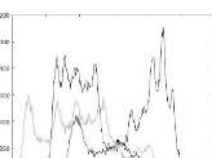
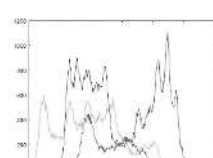
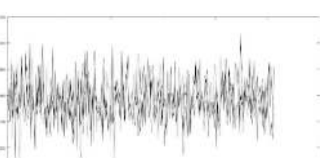
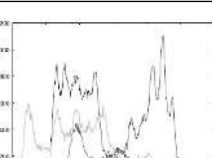
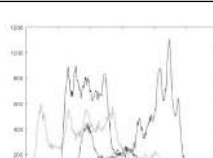
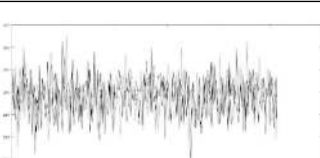
### Histogram Testing

The histogram is a methodology used to calculate the color-level intensity in the image. In the case of gray-level images, the histogram diagram shows how many pixels are each of the 255 grey levels. It can be applied to the color image by separating each channel from 3- dimensional into red, green, and blue histogram individually. The plot is used to compute the number of pixels on y-axis for each color presented on the x-axis. It estimates the distribution of color in original images and encrypted images for image encryption methodology. As shown in the images below, the histogram diagram bars of the images after encryption are evenly distributed among the 255 levels, which is an indication that the quality of the images encrypted by the image encryption methodology is sound.


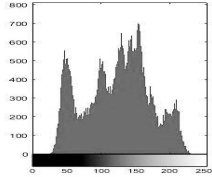
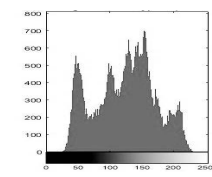
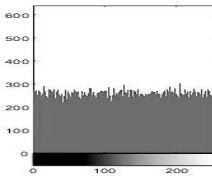

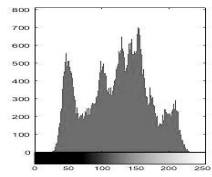
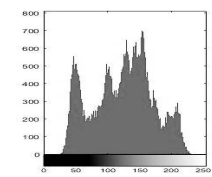
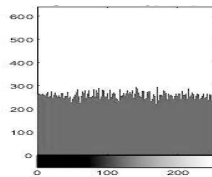

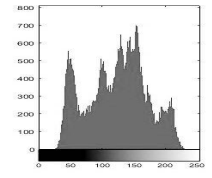
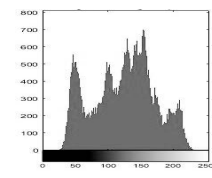
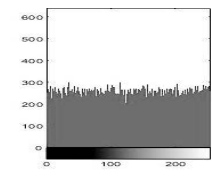

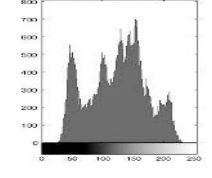
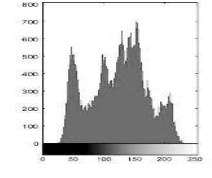
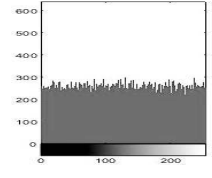

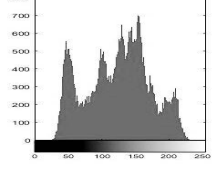
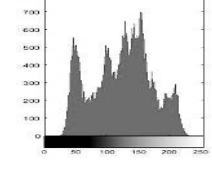
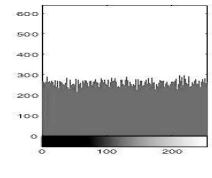

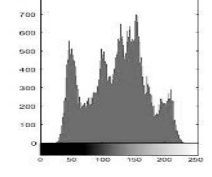
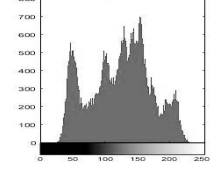
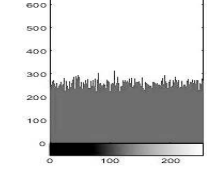
We work through our proposal to implement the algorithm to encrypt images with no statistical link between the original image and the encrypted image. We use the histogram functions to be applied to the grayscale images and color images to view the pixel intensity by graphically displaying the images' pixel distribution before and after the encryption. Table 3 and Table 4 are shown differently after implementing each of the study's cryptographic stages for two samples of grayscale images and color images.

The results are good in all cases of images tests. The difference in pixel positions distribution for all images has been reflected in the pixel shuffled level. Also, the apparent difference in flattening the image's histogram means a change in the pixel values resulting from XORed images. The original images have mostly transparent distribution of pixels in the graph, allowing intruders to be expected to read image information. While encrypted images distribute pixels in the graph, they are uniform and flat, which means that unauthorized can hardly expect image information. The strength of resistance to the statistical attack is enough. The randomness resulting from the distribution of histogram through applying our algorithm ensures good effect and high confidentiality. The algorithm used to encrypt images has achieved our goal of destroying images. It is a safe and efficient way to secure images to hide their features and structures.

**Table 3.** Histogram of color image (Lena.TIF).

Histogram of original image	Histogram of scrambled image	Histogram of ciphered image
<b>Using only Mersenne Twister algorithm</b>		
		
<b>Using only SIMD-oriented Fast Mersenne Twister algorithm</b>		
		
<b>Using only Combined Multiple Recursive algorithm</b>		
		
<b>Using only Multiplicative Lagged Fibonacci Generator algorithm</b>		
		
<b>Using only Legacy MATLAB 5.0 normal generator algorithms</b>		
		
<b>Using only Modified subtract with borrow generator</b>		
		

**Table 4.** Histogram of grayscale image (Lena.TIF).

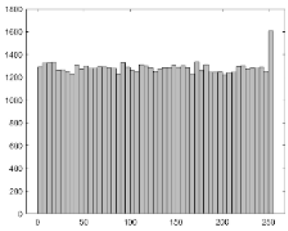
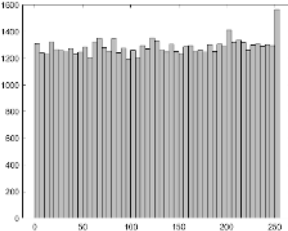
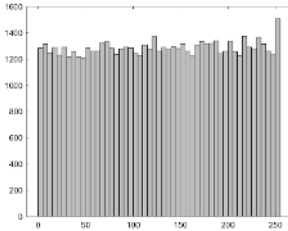
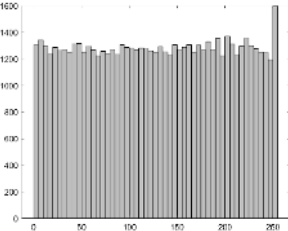
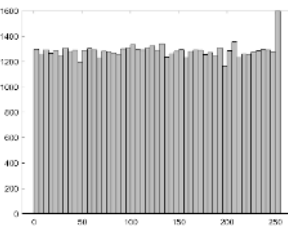
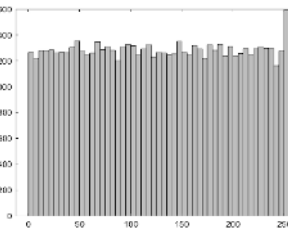
Original image	Histogram of original image	Histogram after scrambled image	Histogram after encrypted image
<b>Mersenne Twister algorithm (twister)</b>			
			
<b>SIMD-oriented Fast Mersenne Twister algorithm (simdTwister)</b>			
			
<b>Combined Multiple Recursive algorithm (combRecursive)</b>			
			
<b>Multiplicative Lagged Fibonacci Generator algorithm (multFibonacci)</b>			
			
<b>Legacy MATLAB 5.0 normal generator algorithms (v5normal)</b>			
			
<b>Modified subtract with borrow generator (swb2712)</b>			
			

### Histogram XOR keys

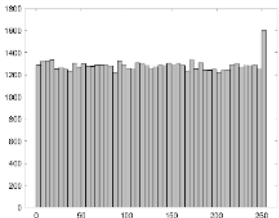
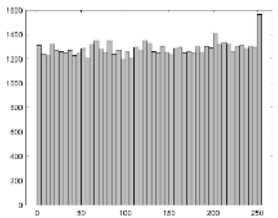
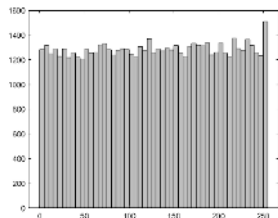
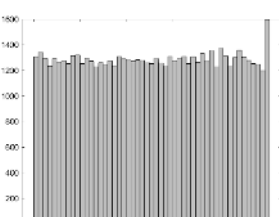
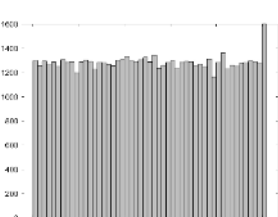
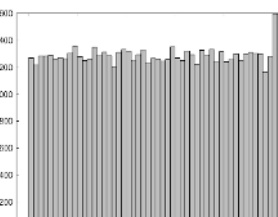
Each time the original image is selected, it is encrypted by two layers of encryption. In the XOR encryption layer, the one-time-pad (OTP) method is used. The power of the stream cipher here is like the power of the random number generation algorithm. In this method, each time the user determines image encryption, an entirely different key is generated from the previous key to encrypt the image. This method prevents hackers and intruders from expecting the key, and it is considered a challenging method to break the key. The key's advantages are that the lowest value is zero, and the highest value is 255, with a specified range (0-255). This allows the determination of the range to be increased, providing effective key strength.

It is challenging to predict key values as the length of the period each time it stands at 256 numbers repeats along the specified repeat period. The period is repeated until the key size is the same as the image to be encrypted in length and width. Keys are generated through 6 generators to generate random numbers. In this study, PRNG generators depend on the initial values to generate a series of random numbers. In this study, the initial value is determined by the pin code from the user. Key values vary based on four main factors: 1/pin code, 2/ Image size, 3/generator used from different 6 PRNG, and 4/the state of the PRNG period within  $2^8-1=255$ . Randomness will be analyzed in all the keys of the 6 PRNG, and the distribution of values will be analyzed through histogram. Table 5 and Table 6 show the distribution of the six generators keys, respectively: 1-Mersenne Twister algorithm, 2-SIMD-oriented Fast Mersenne Twister algorithm, 3-Multiple Multiple Recursive algorithm, 4-Multiplicative Lagged Fibonacci Generator algorithm, 5-Legacy MATLAB 5.0 normal generator algorithms, and 6-Modified subtract with borrow generator.

**Table 5.** Histogram of PRNG keys used for XORing gray image.

Key generated by the Mersenne Twister algorithm	Key generated by the SIMD-oriented Fast Mersenne Twister algorithm	Key generated by the Multiple Recursive algorithm
		
Key generated by the Multiplicative Lagged Fibonacci algorithm	Key generated by the Legacy MATLAB 5.0 normal generator algorithm	Key generated by the Modified subtract with borrow algorithm
		

**Table 6.** Histogram of PRNG keys used for XORing for color image.

Key generated by the Mersenne Twister algorithm	Key generated by the SIMD-oriented Fast Mersenne Twister algorithm	Key generated by the Multiple Recursive algorithm
		
Key generated by the Multiplicative Lagged Fibonacci algorithm	Key generated by the Legacy MATLAB 5.0 normal generator algorithm	Key generated by the Modified subtract with borrow algorithm
		

### Entropy Testing

Entropy is one of the vital image analysis tools in reading information extracted from images. In image analysis, we use a histogram to calculate the number of pixels in images and estimate the probability of distributing each pixel's intensity in color levels. Entropy can measure information in images, whether in a single part of the images or in the entire contents of the image. When computing the histogram of the original image and the encrypted image, the number of different pixels between the histogram computing of the two images appears. The change will be noticeable when the pixel values or pixel positions are changed. Using encryption such as XOR or scrambling encryption helps hide information and maintains randomness. It reflects the randomness of the data, and the satisfactory evaluation is 8.

Each study has to obtain the ideal value 8. The highest number of entropy value indicates the image encrypted. The main idea behind the entropy is the summation of all the possible occurrences of probability distribution pixels, as shown in the following Equation (1) (Zhu et al., 2018).

$$H(I) = - \sum_{i=0}^{2^n-1} p(I)_i \cdot \log_2[p(I)_i] \tag{1}$$

Table 7 and Table 8 show the measuring process of data predictability or the information predictability from the ciphered image into two samples of images grayscale images and color images at two stages of encryption. The first stage is the scrambling operations, and the second stage is XOR encryption. The results we got from the entropy computation was very close to 8. This indicates our algorithm's strength in resisting entropy attacks, and the results confirm the good performance.

The procedures exist within results in Table 7 that appears the best entropy results we have obtained after the image is encrypted with two operations of encryption, with results ranging from 7.9975 to 7.9969. By focusing on the results of the gray image, the result of the entropy indicates that the PRNG that achieved the highest number is

SIMD-oriented Fast Mersenne Twister of 7.9975, and the PRNG that achieved the lowest score of all generators is the generator Legacy MATLAB 5.0 normal and Modified subtract with borrow with equivalent values of 7.9969.

**Table 7.** Entropy of grayscale images.

Original sample image: Lena.tiff	PRNG	The first operation encryption by scrambling image	The second operation encryption by encrypting image with XOR method
7.4429	Mersenne Twister	7.4429	7.9971
	SIMD-oriented Fast Mersenne Twister	7.4429	7.9975
	Multiple Recursive	7.4429	7.9970
	Multiplicative Lagged Fibonacci	7.4429	7.9970
	Legacy MATLAB generator	7.4429	7.9969
	Modified subtract-borrow	7.4429	7.9969

To interpret the performance of the processes in the entropy measurement in Table 8, the results we obtained to test the Lina color image after implementing two encryption operations were between periods 7.9975 and 7.9969. The assessment showed that the generator with the highest value in entropy results is multiplicative Lagged Fibonacci with a value of 7.9975 and the lowest of all generators is the SIMD oriented Fast Mersenne Twister with a value of 7.9969.

**Table 8.** Entropy of RGB images.

Original sample image: Lena.tiff	PRNG	The first operation encryption by scrambled image	The second operation encryption by encrypted image with XOR method
7.7301	Mersenne Twister	7.2404	7.9973
	SIMD-oriented Fast Mersenne Twister	7.2404	7.9969
	Multiple Recursive	7.2404	7.9971
	Multiplicative Lagged Fibonacci	7.2404	7.9975
	Legacy MATLAB generator	7.2404	7.9972
	Modified subtract-borrow	7.2404	7.9971

### Correlation coefficient

The correlation coefficient image test is implemented in order to examine the relationship of the original image with encrypted image. In this test, the original image pixels are measured with the pixels of the encrypted images, similar in principle to the RGB pixel indicator security (Gutub, 2010). Each pixel is measured equally to the corresponding pixels from the encrypted image. Then the resulting number from this test indicates that the pixels are similar or different. The concept of computing the correlation coefficient is based on the following Equation (2), as shown below:

$$r = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{\sum_m \sum_n (A_{mn} - \bar{A})^2 (\sum_m \sum_n (B_{mn} - \bar{B})^2)}} \quad (2)$$

Calculating the correlation coefficient between the pixels and the corresponding pixels between the original and the encrypted images in the gray and color images is done through the following concepts. The pixel value in the original image is calculated compared to the pixel value in the encrypted image that holds the same location for the original image pixels. The rest of the pixels in the two images are compared in the same way as Yen's work (Yen et al., 1996).

The results listed in Table 9 and Table 10 illustrate the computing of the correlation coefficient for the two samples of grayscale image and color image, i.e., with two encryption operations. The results show that our algorithm at both encryptions is very close to 0. This means that the similarity between the original image and the encrypted image is nonexistent.

To clarify the correlation coefficient results from Table 9, the results of the processes we obtained to test the Lina gray image after implementing two encryption operations were between interval 0.0026 and -0.0049. The generator with the highest value close to zero is Legacy MATLAB 5.0 normal with a value of -4.9958e-04, and the lowest generator of all generators that have achieved a value beyond zero is the Multiple Recursive with a value of -0.0049.

**Table 9.** The correlation coefficient of the grayscale image.

A sample study of image: Lena.tif	The first operation encryption by scrambling image	The second operation encryption by XORing image
Mersenne Twister	6.1004e-04	-0.0018
SIMD-oriented Fast Mersenne Twister	-0.0036	-0.0033
Multiple Recursive	2.5615e-04	-0.0049
Multiplicative Lagged Fibonacci	0.0036	0.0016
Legacy MATLAB generator	0.0031	-4.9958e-04
Modified subtract-borrow	0.0058	0.0026

To analyze the correlation coefficient results from Table 10, the results we obtained from the test to investigate the Lina color image after implementing two encryption operations were between interval 0.0033 and -0.0068. The generator with the highest value close to zero is Legacy MATLAB 5.0 normal with a value of -7.7677e-05, and the lowest generator of all generators that have achieved a value beyond zero is the Multiple Recursive with a value of -0.0068.


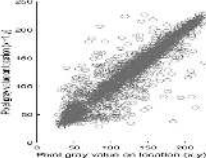
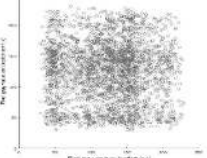
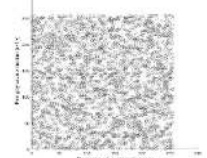

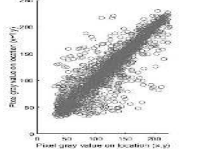
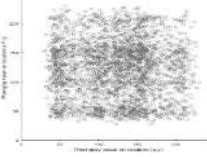
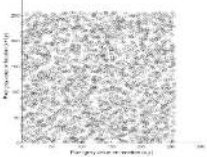

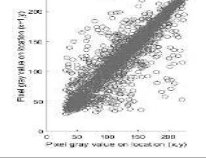
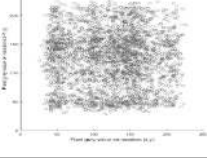
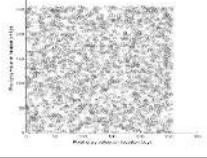

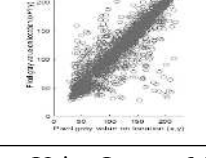
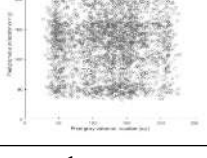
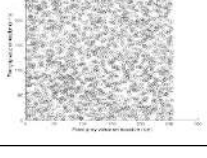

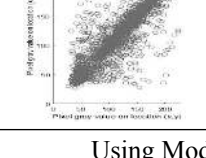
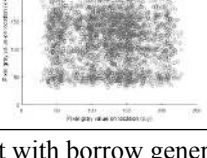
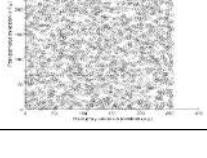

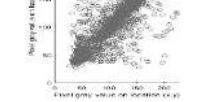
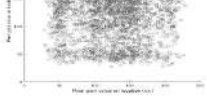

**Table 10.** The correlation coefficient of the color image.

A sample study of image: Lena.tif	The first operation encryption by scrambling image	The second operation encryption by XORing image
Mersenne Twister	-2.3037e-04	-0.0027
SIMD-oriented Fast Mersenne Twister	-0.0029	-0.0068
Multiple Recursive	-6.4147e-04	-0.0019
Multiplicative Lagged Fibonacci	0.0020	0.0033
Legacy MATLAB generator	0.0033	-7.7677e-05
Modified subtract-borrow	0.0053	7.2329e-04

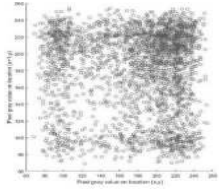
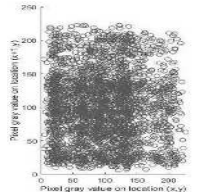
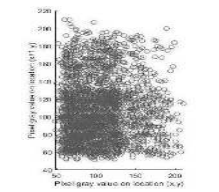
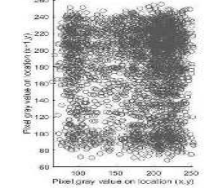
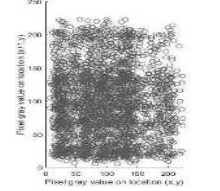
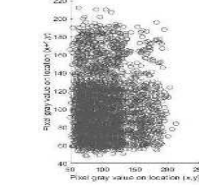
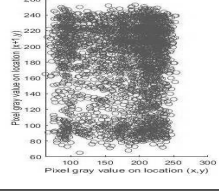
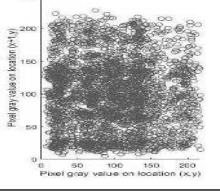
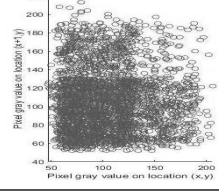
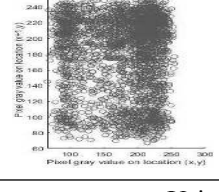
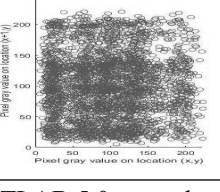
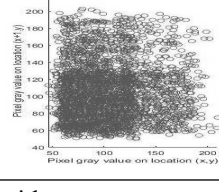
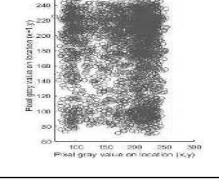
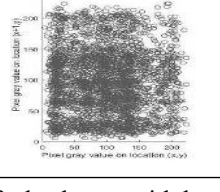
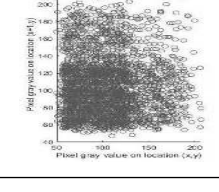
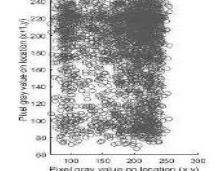
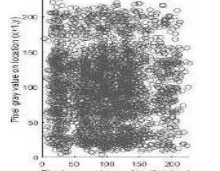
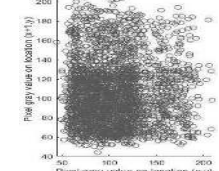


Table 11 shows the correlation in a horizontal orientation for original gray image and its cipher image after two encryption operations. Table 12 and Table 13 present the horizontal orientation of correlation for original color image and its cipher image according to two encryption operations.

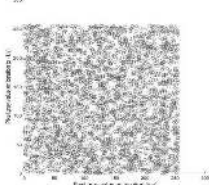
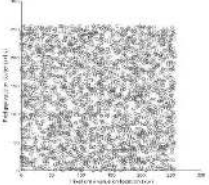
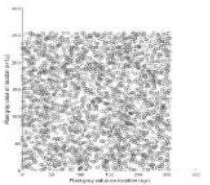
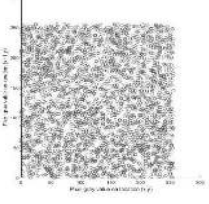
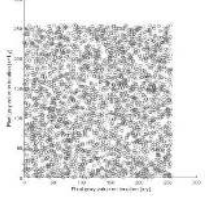
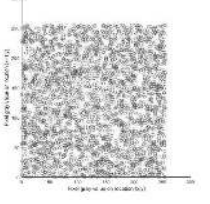
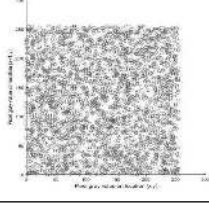
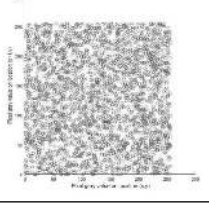
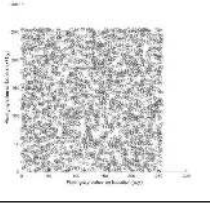
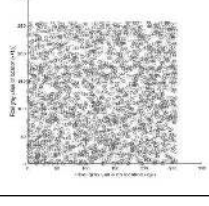
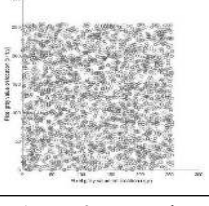
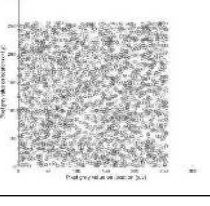
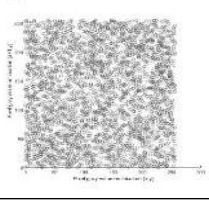
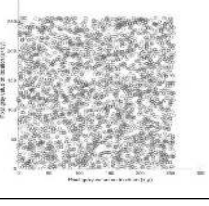
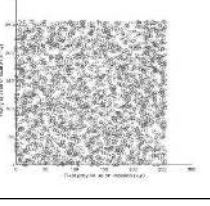
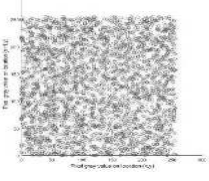
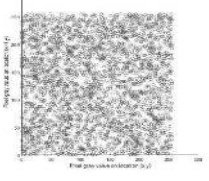
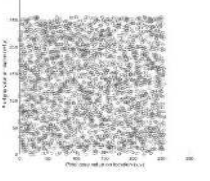
**Table 11.** Horizontal correlation analysis of the grayscale image.

Input image	Correlation corresponding original image	Correlation corresponding scrambled image	Correlation corresponding Ciphred image
<b>Using Merseune Twister algorithm</b>			
			
<b>Using SIMD-oriented Fast Merseune Twister algorithm</b>			
			
<b>Using Combined Multiple Recursive algorithm</b>			
			
<b>Using Multiplicative Lagged Fibonacci Generator algorithm</b>			
			
<b>Using Legacy MATLAB 5.0 normal generator algorithms</b>			
			
<b>Using Modified subtract with borrow generator</b>			
			

**Table 12.** Horizontal Correlation analysis of color image after permutation encryption.

Correlation corresponding Ciphred R channel	Correlation corresponding Ciphred G channel	Correlation corresponding Ciphred B channel
Using Mersenne Twister algorithm		
		
Using SIMD-oriented Fast Mersenne Twister algorithm		
		
Using Combined Multiple Recursive algorithm		
		
Using Multiplicative Lagged Fibonacci Generator algorithm		
		
Using Legacy MATLAB 5.0 normal generator algorithms		
		
Using Modified subtract with borrow generator		
		

**Table 13.** Horizontal Correlation analysis of color image after XOR encryption.

Correlation corresponding Ciphred R channel	Correlation corresponding Ciphred G channel	Correlation corresponding Ciphred B channel
Using Mersenne Twister algorithm		
		
Using SIMD-oriented Fast Mersenne Twister algorithm		
		
Using Combined Multiple Recursive algorithm		
		
Using Multiplicative Lagged Fibonacci Generator algorithm		
		
Using Legacy MATLAB 5.0 normal generator algorithms		
		
Using Modified subtract with borrow generator		
		

### Image Quality Measurements

We can determine the image quality by studying the following three measures: Mean Square Error (MSE), Peak Signal to noise ratio (PSNR), structural similarity index (SSIM), and Mutual Information (MI). If the MSE result is studied and the value appears to be zero, the highest value of the two images' similarity has been achieved. The opposite represents the dissimilarity of the two images. The MSE displays the error between the corresponding pixels of two different images, the error between the original image and our encrypted image. The error value determines the difference between the original image from the encrypted image.

The PSNR is defined as the peak signal-to-noise ratio between two images, i.e., the original images and encrypted images. The highest value in PSNR means the images' quality is good, and the lower value means the whole image is noise. Another quality measurement is SSIM. It takes a value between zero and one. The closer the value is to one, or it becomes one, the image quality is good, and the quality image is similar to the original image. The high value in SSIM means that the two comparative images are the same. The opposite value means that the two images are totally different. The last measurement will be about how closely the two images relate. MI measure this. The value of MI indicates the extent to which the encrypted image contains information about the original image. The following Equations (3), (4), (5), and(6) show the mathematical calculation of all the quality measures of the images.

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \quad (3)$$

$$PSNR = 10 \log_{10} \left( \frac{255^2}{MSE} \right) \quad (4)$$

$$SSIM = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (5)$$

$$M(X; Y) = \sum_x \sum_y p(x, y) \log \left( \frac{p(x, y)}{p(x)p(y)} \right) \quad (6)$$

Table 14 and Table 15 show the results of quality measures for gray and color images at both encryption operations. All metrics applied between the original images with the encrypted image. The result of Peak signal to noise ratio (PSNR) is low as should be for totally dissimilar images. Mean square error (MSE) is very high, and cross correlation is low. Structural similarity index (SSIM) is nearly 0 for all images, which means that the similarity between images and their encrypted versions is nonexistent. The Mutual information (MI) measure is also close to 0, which means that there is no mutual information between an image and its encrypted version. Figure 8 and Figure 9 show combined results of different evaluation metrics of grey and color image quality measures after two encryption operations.

Table 14 shows four metrics that test the encryption applied to the Lina gray image after our two operations of encryption. The results of the operations we obtained from the MSE measurement falls between the periods 7.8276e+03 and 7.7153e+03. The generator with the highest value is the Multiple Recursive generator with a value of 7.8276e+03, and the lowest generator of all generators is the Multiplicative Lagged Fibonacci with a value of 7.7153e+03. The results of the operations we obtained from the PSNR measurement were between periods 9.1945 and 9.2573. The generator that received the lowest value from PSNR is the Multiple Recursive generator with a value of 9.1945, and the highest generator of all generators that achieved the highest value of PSNR is the Multiplicative Lagged Fibonacci with a value of 9.2573. By observing the PSNR results and MSE, we noted that the preference is equal in terms of generators due to the dependability of PSNR in its calculation on the result of the MSE. Also, good results state that the SSIM measurement results fall between the periods 0.0079 and 0.0113. The highest efficiency generator in SSIM-

oriented Fast Mersenne appeared at a value close to zero of 0.0079, and the lowest efficiency of the Multiplicative Lagged Fibonacci generator appeared at a value of 0.0113. MI results confirm that the SSIM-oriented Fast Mersenne generator is the highest in terms of a value close to zero with a value equal to 0.5073, and multiplicative Lagged Fibonacci produced the most distant score of zero with a value equal to 0.5163.

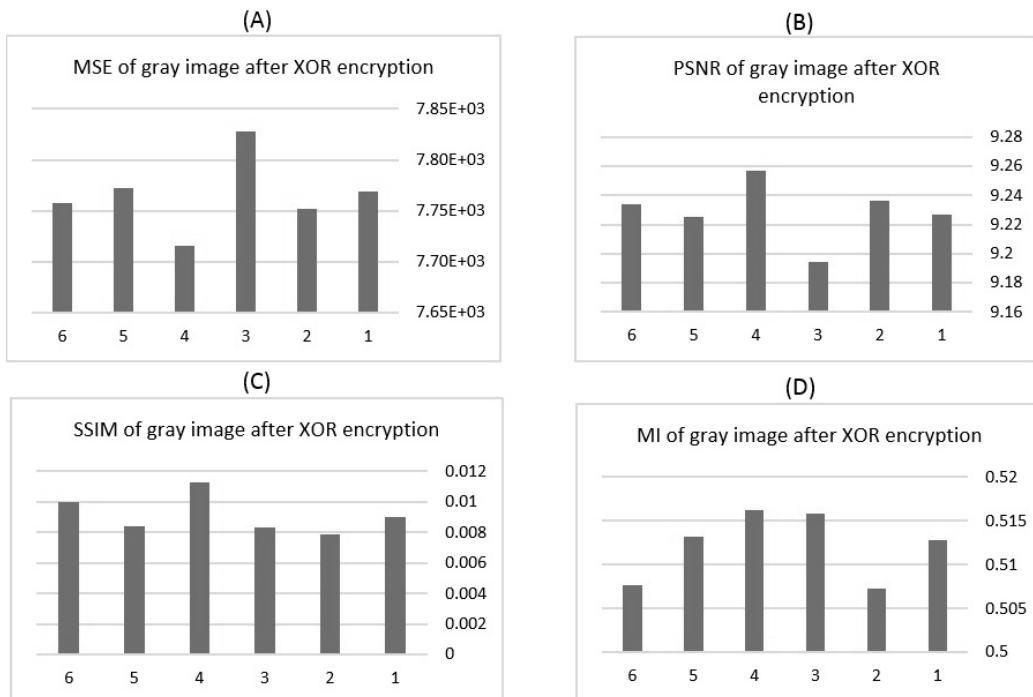
**Table 14.** Quality image of the grayscale image.

The first operation encryption by scrambled image				The second operation encryption by encrypted image			
MSE	PSNR	SSIM	MI	MSE	PSNR	SSIM	MI
Using Mersenne Twister algorithm							
4.5742e+03	11.5277	0.0240	0.3170	7.7690e+03	9.2271	0.0090	0.5128
Using SIMD-oriented Fast Mersenne Twister algorithm							
4.5936e+03	11.5092	0.0212	0.3188	7.7519e+03	9.2367	0.0079	0.5073
Using the Multiple Recursive algorithm							
4.5758e+03	11.5261	0.0249	0.3164	7.8276e+03	9.1945	0.0083	0.5158
Using the Multiplicative Lagged Fibonacci algorithm							
4.5605e+03	11.5407	0.0245	0.3173	7.7153e+03	9.2573	0.0113	0.5163
Using the Legacy MATLAB 5.0 normal generator algorithm							
4.5629e+03	11.5384	0.0212	0.3197	7.7720e+03	9.2255	0.0084	0.5132
Using the Modified subtract with borrow algorithm							
4.5503e+03	11.5504	0.0232	0.3180	7.7575e+03	9.2336	0.0100	0.5077

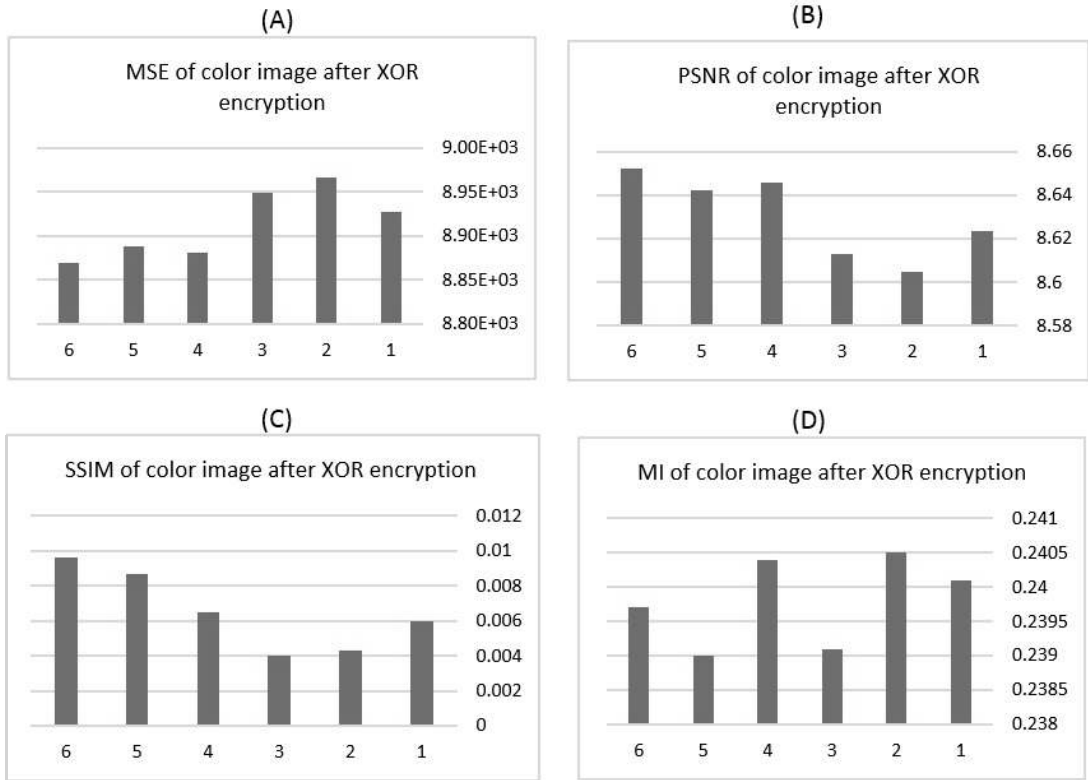
The results of Table 15 indicate four metrics applied to Lina color image after XOR encryption. We obtained from the MSE measurement falls between the periods 8.9662e+03 and 8.8689e+03. The generator with the highest is SIMD-oriented Fast Mersenne Twister with a value of 8.9662e+03, and the lowest generator of all generators is the Modified subtract with borrow with a value of 8.8689e+03. The results of the operations we obtained from the PSNR measurement were between periods 8.6047 and 8.6521. The generator that received the lowest value from PSNR is the SIMD-oriented Fast Mersenne Twister with a value of 8.6047 and the highest generator of all generators that achieved the highest value of PSNR is the Modified subtract with borrow with a value of 8.6521. Also, good results state that the results of the SSIM measurement fall between the periods 0.0040 and 0.0096. The highest-efficiency generator in Multiple Recursive appeared at a value close to zero of 0.0040 and the lowest Modified subtract with borrow generator appeared at a value of 0.0096. MI results confirm that the Legacy MATLAB 5.0 normal generator is the highest in terms of a value close to zero with a value equal to 0.2390 and SIMD-oriented Fast Mersenne Twister produced the most distant score of zero with a value equal to 0.2405.

**Table 15.** Quality image of the color image.

The first operation encryption by scrambled image				The second operation encryption by encrypted image			
MSE	PSNR	SSIM	MI	MSE	PSNR	SSIM	MI
Using only Mersenne Twister algorithm							
4.1415e+03	11.9592	0.4726	0.3725	8.9271e+03	8.6237	0.0060	0.2401
Using only SIMD-oriented Fast Mersenne Twister algorithm							
4.1529e+03	11.9473	0.4709	0.3695	8.9662e+03	8.6047	0.0043	0.2405
Using only the Multiple Recursive algorithm							
4.1423e+03	11.9584	0.4734	0.3725	8.9484e+03	8.6133	0.0040	0.2391
Using only the Multiplicative Lagged Fibonacci algorithm							
4.1319e+03	11.9693	0.4730	0.3708	8.8813e+03	8.6460	0.0065	0.2404
Using only the Legacy MATLAB 5.0 normal generator algorithm							
4.1269e+03	11.9745	0.4718	0.3733	8.8889e+03	8.6423	0.0087	0.2390
Using only the Modified subtract with borrow algorithm							
4.1191e+03	11.9828	0.4729	0.3699	8.8689e+03	8.6521	0.0096	0.2397



**Figure 6.** Quality image of grayscale Lina image with two-operation encryption using the six PRNG measurements results: (A) MSE. (B) PSNR. (C) SSIM. (D) MI.



**Figure 7.** Quality image of color Lina image with two-operation encryption using the six PRNG measurement results: (A) MSE. (B) PSNR. (C) SSIM. (D) MI.

**Sensitive analysis to resist differential attack**

To measure the sensitivity of the change in the original images and the strength of the algorithm used in encryption. We used two of the metrics that make it happen: the number of pixels changing rate (NPCR), and the unified averaged changed intensity (UACI). We can define the following Equations:

$$NPCR = \frac{1}{m \times n} \sum_{i=1}^m \sum_{j=1}^n \delta(i, j) \times 100\% \tag{7}$$

$$UACI = \frac{1}{m \times n} \left( \sum_{i=1}^m \sum_{j=1}^n \frac{|C_1(i, j) - C_2(i, j)|}{255} \right) \times 100\% \tag{8}$$

These metrics are sensitive to any change in images, even if the change is a single pixel that notices the tiny change (Wu et al., 2011). The NPCR calculates and measures the percentage of pixel change in encrypted images, and this value means the change in the pixel of original image. The UACI measures the average pixel intensity in encrypted images, and this value means the change in pixel intensity in the original image. In our algorithm, all sensitivity metrics have achieved close to ideal values, and this increase and proves the strength of the algorithm used in image encryption. Table 16 and Table 17 show the results of gray and color images at both encryption operations. The UACI should be close to 33%, and the theoretical NPCR is between 99.5810 and 99.5893. The two metrics achieved good results in the second encryption operation for encrypting gray and color images.

**Table 16.** NPCR and UACI for the gray image.

A sample study of image: Lena.tif	The first operation encryption by scrambling image		The second operation encryption by encrypting image	
PRNG	NPCR%	UACI%	NPCR%	UACI%
Mersenne Twister	99.32	22.66	99.61	28.66
SIMD-oriented Fast Mersenne Twister	99.35	22.67	99.62	28.60
Multiple Recursive	99.29	22.63	99.62	28.77
Multiplicative Lagged Fibonacci	99.34	22.56	99.61	28.51
Legacy MATLAB generator	99.36	22.60	99.61	28.63
Modified subtract-borrow	99.39	21.43	99.64	28.64

**Table 17.** NPCR and UACI for color images.

Sample study of image: Lena.tif	The first operation encryption by scrambling image		The second operation encryption by encrypting image	
PRNG	NPCR	UACI	NPCR	UACI
Mersenne Twister	99.23	19.78	99.62	30.43
SIMD-oriented Fast Mersenne Twister	99.23	19.78	99.61	30.50
Multiple Recursive	99.25	19.77	99.59	30.47
Multiplicative Lagged Fibonacci	99.22	19.73	99.60	30.32
Legacy MATLAB generator	99.28	19.74	99.62	30.33
Modified subtract-borrow	99.25	19.71	99.64	30.31

### Execution Time Analysis

Good encryption processes depend on the flexibility of the algorithms used to encrypt images and processes' speed. So, we analyzed and calculated the time required to execute the program for operations. The purpose of the analysis is to confirm the arrival of our algorithm for flexible and rapid implementation. To test encryption speed, the test was conducted on 256 size, and TIF form of images, different types of images of both gray and colored, and at different operations of encryption. Following Table 18 and Table 19 are the sample images that we have validated the speed of our algorithm.

Table 18 confirms that the high-speed generator of all six PRNG is SIMD-oriented Fast Mersenne Twister with an estimated time of 0.2038, and the lowest speed among them is multiplicative Lagged Fibonacci with an estimated time of 0.2915. Table 19 consists of results of time on Lena color image encryption. Emphasis on the high-speed generator of all six generators is Legacy MATLAB 5.0 normal with a required time of 0.1976 and the lowest speed among them is Modified subtract with borrow with a required time of 0.3045.



**Table 18.** Time cost for gray images.

A sample study of image: Lena.tif	The first operation encryption by scrambled image	The second operation encryption by encrypted image
PRNG	Time (Sec)	Time (Sec)
Mersenne Twister	0.3492	0.2122
SIMD-oriented Fast Mersenne Twister	0.2325	0.2038
Multiple Recursive	0.2820	0.2091
Multiplicative Lagged Fibonacci	0.2449	0.2915
Legacy MATLAB generator	0.2672	0.2174
Modified subtract-borrow	0.2782	0.2243

**Table 19.** Time cost for a color image.

A sample study of image: Lena.tif	The first operation encryption by scrambled image	The second operation encryption by encrypted image
PRNG	Time (Sec)	Time (Sec)
Mersenne Twister	0.1986	0.2762
SIMD-oriented Fast Mersenne Twister	0.2049	0.2078
Multiple Recursive	0.2355	0.2454
Multiplicative Lagged Fibonacci	0.2376	0.2341
Legacy MATLAB generator	0.2051	0.1976
Modified subtract-borrow	0.3617	0.3045

## COMPARISONS AND REMARKS

With the results we have already obtained, we can conclude good results from the six different PRNGs in the encryption of gray and color images. This section will display the effective PRNG that achieved the highest metric performance for the encryption processes and analyze image encryption performance of the two operations (transposition and substitution) in image pixels in terms of confidentiality and indicators of the results of measurements of images after encryption.

### Comparison of Changing PRNG

As shown in Table 20, the first operation achieved significantly lower results than the confidentiality and power of encryption in the second operation of image encryption, i.e., by applying the XOR process that follows the first pixel shuffling. The structural distribution of image pixels appeared in the histogram in flat form, where the image's features were concealed, increasing the value of the entropy with results very close to eight. The value of MSE increased fold after the second operation, which increased the noise in PSNR significantly. The analysis of the following discussions shall focus on the second operation of encryption as a result of the preference achieved in the results, and each generator will be examined individually and as to how effective it is in encrypting gray and colored images.

**Table 20.** Encryption operations show goals of metric results.

operation	High Security	Low entropy	Destroyed images quality	Reduce the correlation
The first operation of encryption		✓		✓
The second operation of image encryption	✓		✓	

The measurements imply that a good generator and the highest performance of gray image encryption is the one that achieves the highest results in more than one measure. The following results are to examine the case of the Lena gray image of a size 256 after the two encryptions operations, as shown in Table 21. Of the six random generators that achieved preference in performance, the SIMD-oriented Fast Mersenne Twister generator achieved the highest in four of the seven metrics: entropy, SSIM, MI, and time. Therefore, the SIMD-oriented Fast Mersenne Twister generator is suitable to encrypt the gray Lina image. On the other side, the generator that achieved the lowest results in all measures is Multiplicative Lagged Fibonacci generator. The result remarks poor performance compared to other generators used in encryption by focusing on the generator performance indicator as reached low in five of the seven measures.

**Table 21.** Measurements comparison testing for the grayscale image encryption.

PRNG	Entropy	Cross correlation	MSE	PSNR	SSIM	MI	Time(sec)
Mersenne Twister	Moderate	Moderate	Moderate	Moderate	Moderate	Moderate	Moderate
SIMD-oriented Fast Mersenne Twister	High	Moderate	Moderate	Moderate	High	High	High
Multiple Recursive	Moderate	Low	High	High	Moderate	Moderate	Moderate
Multiplicative Lagged Fibonacci	Moderate	Moderate	Low	Low	Low	Low	Low
Legacy MATLAB generator	Low	High	Moderate	Moderate	Moderate	Moderate	Moderate
Modified subtract-borrow	Low	Moderate	Moderate	Moderate	Moderate	Moderate	Moderate

We estimate that the results of high measurements determine singularity in good performance of the generator or not. After conducting performance tests on Lena color image by encrypting it with two encryption operations, we find the following results in Table 22 that shows the six random generators with the highest and worst performance. The Legacy MATLAB 5.0 normal generator achieved the highest in three of the seven metrics: correlation coefficient, MI, and time. Therefore, the Legacy MATLAB 5.0 normal generator is suitable to encrypt the color Lina image. On the other side, the generator that achieved the lowest results in all measures is Modified subtract with borrow generator. By focusing on the generator performance indicator showing the lowest values in four of the seven measures, the result can be claimed as poor performance, i.e., compared to using other generators.

**Table 22.** Measurements comparison testing for the RGB image encryption.

PRNG	Entropy	Cross correlation	MSE	PSNR	SSIM	MI	Time(sec)
Mersenne Twister	Moderate	Moderate	Moderate	Moderate	Moderate	Moderate	Moderate
SIMD-oriented Fast Mersenne Twister	Moderate	Low	High	High	Moderate	Low	Moderate
Multiple Recursive	Moderate	Moderate	Moderate	Moderate	High	Moderate	Moderate
Multiplicative Lagged Fibonacci	High	Moderate	Moderate	Moderate	Moderate	Moderate	Moderate
Legacy MATLAB generator	Moderate	High	Moderate	Moderate	Moderate	High	High
Modified subtract-borrow	Moderate	Moderate	Low	Low	Low	low	Low

### Comparison to other studies

We analyze the performance of encryption processes while comparing them with the rest of the studies and see that the analysis of the above results shows the typical measurement of ciphered image.

After two encryption procedures, the images we used in our algorithm have achieved the required number in the entropy to be close to the eight results, which we were seeking. This means that encrypted images are sufficient to be randomly from their source, which means that our encryption operations are secure against an entropy attack. The following Table 23 shows the results compared to the rest of the studies. The results we got are from 7.9975 to 7.9969. The highest entropy we got is 9.9975; the generator we used with encryption is SIMD-oriented Fast Mersenne Twister algorithm, which is higher in terms of result than the other two studies.

**Table 23.** Entropy comparison of the results to other schemes.

PRNG	Our Proposal	(Banthia and Tiwari, 2013)		(Rohith et al., 2014)	
		LCG	Logistic map	Logistic map	Logistic map & LFSR
Gray Lena.tif 256*256					
Mersenne Twister	7.9971	7.9533 to 7.9665	7.8837 to 7.9325	7.9737	7.9974
SIMD-oriented Fast Mersenne Twister	7.9975				
Multiple Recursive	7.9970				
Multiplicative Lagged Fibonacci	7.9970				
Legacy MATLAB generator	7.9969				
Modified subtract-borrow	7.9969				

Looking at the relationship between the original and encrypted images, the other studies seek to reduce this relationship through the optimal encryption processes that make the relationship of the encrypted image with the original image entirely different and with no correlation. Table 24 describes our results with other studies that show that the relationship between the two images is too low.

The results obtained are representing that three generators used in encryption gave better results than other studies. In Banthia and Tiwari (2013), the lowest value by LCG is 0.0019 compared with our result, which shows the three generators of our study as better affectivity. The following three values are 0.0016, -0.0018, and -4.9958e-04 smaller correlation are the values we got compared to the other study in Banthia and Tiwari (2013) that used the LCG generator, while the value of the same study using Logistic map 9.72E-05 was the smallest compared to the results we received. When comparing our results with the second study (Rohith et al., 2014), all the generators we used gave the smallest values compared to the Logistic map method. Four of our generators gave the smallest values compared to the second improved method in (Rohith et al., 2014). The following best results are -0.0018, 0.0016, 0.0026, and -4.9958e-04, which are smaller and closer to zero, i.e., than the 0.0036 value.

**Table 24.** Correlation comparison of the results to other schemes.

PRNG	Our Proposal	(Banthia and Tiwari, 2013)		(Rohith et al., 2014)	
		LCG	Logistic map	Logistic map	Logistic map & LFSR
	Lena image 512x512				
Mersenne Twister	-0.0018	- 0.01 to 0.01	- 0.006 to 0.0036	-0.0133	0.0036
SIMD-oriented Fast Mersenne Twister	-0.0033				
Multiple Recursive	-0.0049				
Multiplicative Lagged Fibonacci	0.0016				
Legacy MATLAB generator	-4.99e-04				
Modified subtract-borrow	0.0026				

In other studies, image quality measurements vary in results. High MSE measurements demonstrate how powerful the algorithm is in making the two images different with distinct measurements. The result of the PSNR of encrypted images is that the result is as low as possible because it indicates that the image is destroyed and information cannot be read. Table 25 shows the differences in results with other studies. In the (Banthia and Tiwari, 2013) study, the LCG method obtained several readings of the MSE and PSNR results; the highest detected result for MSE is 7.43E03 and for PSNR, it is 9.4206. Comparing the LCG result with our results suggests that our four generators gave better results in measuring MSE and PSNR values. The results of the MSE measurement respectively are 7.8276e+03, 7.7720e+03, 7.7690e+03, 7.7575e+03 and the results of the PSNR measurement respectively are 9.1945, 9.2255, 9.2271, 9.2336 and 9.2573 better and higher than MSE value 7.43E03 and PSNR value 9.4206. The Banthia and Tiwari (2013) study also shows the results of the Logistic map method after taking several readings to measure MSE, PSNR for Lena gray encrypted image reached the highest result in the MSE value of 9.16E+03 and the lowest value in the PSNR measurement of 8.5112 values that were monitored better than our results, while, in the (Rohith et al., 2014) study, the results of the two methods used in encryption in measuring MSE are much lower than the results we obtained. This means that our results in the six generators compared with the (Rohith et al., 2014) study in the MSE are higher.

**Table 25.** image quality comparison of the results to other schemes.

PRNG	Our system		(Banthia and Tiwari, 2013)				(Rohith et al., 2014)	
	MSE	PSNR	LCG	Logistic map	LCG	Logistic map	Logistic map	map & LFSR
			MSE		PSNR		MSE	
	Gray Lena image 512x512							
Mersenne Twister	7.7690e+03	9.2271						
SIMD-oriented Fast Mersenne Twister	7.7519e+03	9.2367						
Multiple Recursive	7.8276e+03	9.1945	6.80E+03	8.87E+03	9.42	8.51	7095.9	7591.5
Multiplicative Lagged Fibonacci	7.7153e+03	9.2573	to 7.43E+03	to 9.16E+03	to 9.80	to 8.65		
Legacy MATLAB generator	7.7720e+03	9.2255						
Modified subtract-borrow	7.7575e+03	9.2336						

## CONCLUSION

With the internet spreading widely, verifying images turned into a need in numerous fields, for example, the therapeutic field, the military field, and numerous other significant fields. The paper's fundamental point is to introduce a secure system for private images transmission over the web. This paper shall give the experimentation subtleties with the resulted outcomes, indicating that PRNG is a suitable, solid, and productive calculation for encrypting and decrypting images. The contribution lies in combining multiple algorithms acting as the pseudorandom number generators that will be used to decrypt images based on a key given by the user.

This paper provides an image encryption methodology with robust encryption methodology. The first encryption PRNG operation generates numbers of sequences used for shuffling each pixel's positions to produce an encrypted image with the details being totally wiped. At the second operation of encryption, keys are generated for the XOR operations.

The performance of the PRNG based image encryption methodology was evaluated using statistical measurements such as histogram, entropy, correlation coefficient, MSE, PSNR, SSIM, MI, NPCR, UACI, and speed. The experimentation results have shown that the PRNG-based image encryption methodology complied with the three-security standards integrity, authentication, and confidentiality. Image results measurements achieved the required standard values. The results of the two-operation transposition encryption, followed by XOR encryption, respectively, showed high results. The entropy results are close to 8, where one of the images achieved a score of 7.9975, which is the highest value for encrypting gray and color images, while the cross-correlation results of all the images were close to 0, and the highest score we recorded is  $-7.7677e-05$  between the original color image and their encrypted image. All image quality measurements achieved the rates we were looking for. The MSE results are high and confirm the encrypted image's difference from the original image with high results. The PSNR emphasizes the noise of the fully encrypted image in low-quality measures. The similarity of the images and their relationship to each other and the extent to which the encrypted and original images is containing information as low as they should be in the SSIM and MI measure is discussed. The number of pixels change rate (NPCR) is 100% for all images, which means that every pixel changed value. Unified Average Changing Intensity (UACI) is also extremely low.

As a future work plan, this paper's study proposes to study all cases of image encryption compared with gray and colored samples with different methodology. This is to show the differences that can occur and study them in terms of changing the order of the encryption method. Instead of relying on the first operation of image encryption as transposition, we will switch to make the first operation as XOR substitution, making the second operation as transposition. We can apply all the security measurements studied in this paper to compare the results with them, assuming the cost is the same, showing more attractive remarks.

## ACKNOWLEDGMENT

This work has been supported by Umm Al-Qura University. The authors thanks the College of Computer and Information System for providing assistance to make this research possible.

## REFERENCES

- Al-Juaid, N., Gutub, A. & Khan, E. 2018.** Enhancing PC data security via combining RSA cryptography and video based steganography. *Journal of Information Security and Cybercrimes Research (JISCR)*, **1**(1): 8-18
- Al-Juaid, N. & Gutub, A. 2019.** Combining RSA and audio steganography on personal computers for enhancing security. *SN Applied Sciences* **1**:830.
- Al-Otaibi, N. & Gutub, A. 2014.** 2-layer security system for hiding sensitive text data on personal computers. *Lecture Notes on Information Theory*, **2**(2): 151-157.
- Alharthi, N. & Gutub, A. 2017.** Data visualization to explore improving decision-making within Hajj services. *Scientific Modelling and Research*, **2**(1): 9-18.
- Almutairi, S., Gutub, A. & Al-Juaid, N. 2020.** Motivating Teachers to Use Information Technology in Educational Process within Saudi Arabia. *International Journal of Technology Enhanced Learning (IJTEL)*, **12**(2): 200-217.
- Almutairi, S., Gutub, A. & Al-Ghamdi, M. 2019.** Image Steganography to Facilitate Online Students Account System. *Review of Business and Technology Research (RBTR)*, **16**(2): 43-49
- Banthia, A. & Tiwari, N. 2013.** Image Encryption using Pseudo Random Number Generators. *International Journal of Computer Applications*, **975**: 8887.
- Bin-Hureib, E. & Gutub, A. 2020.** Enhancing Medical Data Security via Combining Elliptic Curve Cryptography and Image Steganography. *International Journal of Computer Science and Network Security (IJCSNS)*, **20**(8): 1-8.
- El-Samie, F., Ahmed, H., Elashry, I., Shahieen, M., Faragallah, O., El-Rabaie, E. & Alshebeili, S. 2013.** *Image encryption: a communication perspective*, CRC Press.
- Gutub, A. & Al-Qurashi, A. 2020.** Secure Shares Generation via M-Blocks Partitioning for Counting-Based Secret Sharing. *Journal of Engineering Research*, **8**(3): 91-117.
- Gutub, A., Al-Juaid, N. & Khan, E. 2019.** Counting-based secret sharing technique for multimedia applications. *Multimedia Tools and Applications*, **78**: 5591-5619.
- Gutub, A. 2011.** Subthreshold SRAM designs for cryptography security computations. *International Conference on Software Engineering and Computer Systems*, pp. 104-110.
- Gutub, A. 2010.** Pixel Indicator Technique for RGB Image Steganography. *Journal of Emerging Technologies in Web Intelligence (JETWI)*, **2**(1): 56-64.
- Gutub, A. & Fattani, M. 2007.** A Novel Arabic Text Steganography Method Using Letter Points and Extensions. *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, **1**(3): 502-505.
- Gutub, A. & Tenca, A.F. 2004.** Efficient scalable VLSI architecture for Montgomery inversion in GF(p). *Integration, the VLSI journal*, **37**(2): 103-120.
- Hassan, F. & Gutub, A. 2020.** Efficient Reversible Data Hiding Multimedia Technique Based on Smart Image Interpolation. *Multimedia Tools and Applications*, **79**(39): 30087-30109.

- Kapur, V., Paladi, S. & Dubbakula, N. 2015.** Two level image encryption using pseudo random number generators. *International Journal of Computer Applications*, 115.
- Khshaifaty, N. & Gutub, A. 2020.** Preventing Multiple Accessing Attacks via Efficient Integration of Captcha Crypto Hash Functions. *International Journal of Computer Science and Network Security (IJCSNS)*, **20(9)**: 16-28.
- Ramesh, A. & Jain, A. 2015.** Hybrid image encryption using Pseudo Random Number Generators, and transposition and substitution techniques. *IEEE International Conference on Trends in Automation, Communications and Computing Technology (I-TACT-15)*, pp. 1-6.
- Rohith, S., Bhat, K. & Sharma, A. 2014.** Image encryption and decryption using chaotic key sequence generated by sequence of logistic map and sequence of states of Linear Feedback Shift Register. *IEEE International Conference on Advances in Electronics Computers and Communications*, pp. 1-6.
- Saha, S., Karsh, R. & Amrohi, M. 2018.** Encryption and Decryption of Images Using Secure Linear Feedback Shift Registers. *IEEE International Conference on Communication and Signal Processing (ICCSP)*, pp. 0295-0298.
- Sarma, K. & Lavanya, B. 2017.** Digital image scrambling based on sequence generation. *IEEE International Conference on Circuit, Power and Computing Technologies (ICCPCT)*, pp. 1-5.
- Sharma, M. & Kowar, M. 2010.** Image encryption techniques using chaotic schemes: a review. *International Journal of Engineering Science and Technology*, **2(6)**: 2359-2363.
- Wu, Y., Noonan, J. & Aгаian, S. 2011.** NPCR and UACI randomness tests for image encryption. *Cyber journals: multidisciplinary journals in science and technology, Journal of Selected Areas in Telecommunications (JSAT)*, **1**: 31-38.
- Yen, E. & Johnston, R. 1996.** The ineffectiveness of the correlation coefficient for image comparisons. *Technical Report LA-UR-96-2474*, Los Alamos National Laboratory.
- Zhu, C., Wang, G. & Sun, K. 2018.** Improved cryptanalysis and enhancements of an image encryption scheme using combined 1D chaotic maps. *Entropy*, **20**: 843.