Multimedia Tools and Applications (2021) 80:28521–28581
https://doi.org/10.1007/s11042-021-11051-3

Remodeling randomness prioritization to boost-up security of RGB image encryption

Budoor Obid Al-Roithy[1] · Adnan Gutub[1]

Received: 2 September 2020 / Revised: 7 January 2021 / Accepted: 5 May 2021 /
Published online: 7 June 2021

Abstract
Securing information became essential to exchange multimedia information safely. The exchanged data need to be transformed in a well-managed, secure, and reliable manner. In this paper, we will focus on securing RGB images via cryptography during transmission among users using our effective proposal of utilizing appropriate Pseudo Random Number Generator (PRNG). We implement many techniques of PRNG involved in two consecutive crypto-processes of substitution and transposition to present secure image transformation. Our technique proposal of PRNGs selection is based on testing to encrypt RGB images to be compared with current related used approaches. The work experimentation aims to identify suitability and reliability through security measures standard parameters. The research justifies its proper PRNG selection to model our approach as attractive effective work worth remarking.

Keywords Image encryption · Pseudo random number generator (PRNG) · Digital image · Image scrambling · Image shuffling

1 Introduction

Everyone is turning to the digital world. The Internet considers to be an essential part of communication between all parties. Internet communications are vulnerable to violations in the security triangle which need proper ensuring the security of information. Any data transfers through normal insecure communication channels are subject to be under attack affecting its Confidentiality, Integrity, and Availability (CIA). They are the main threats considered in the field of information security. Securing these three components (CIA) of information security has become an urgent need [32].

✉ Adnan Gutub
aagutub@uqu.edu.sa

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

Today, our living in the digital world puts all communication multimedia to be facing urgent security challenges. Digital media includes several types, such as audio, videos, images, holograms, social media, and any virtual reality multimedia [8]. It exists in many industries like those associated with education, health, and government. This confirms the digital community needs for a continued secure environment that guarantees delivering and exchanging data to the user safely [3].

Nowadays, the usage of images has become more widespread among users. So, encryption data is necessary for protecting the privacy of content and preventing unauthorized control or access to information by modifications. This implies a necessity for verifying the authentication of the image via passwords for their security being vital multimedia data [3].

Currently, users' sensitive images must be protected via cryptography [3]. The use of data encryption is urgently needed to be protected when transmitted through unsafe Internet channels [30]. Cryptography converts data to another form such that it changes texts from plain texts to ciphertexts form [7]. Encryption is the making of many different mathematical calculations to substitute the data and convert it into useless information [31]. It needs a secret key for encrypting and decrypting the data to ensure security [4].

Most applications are involving digital images in secure scenarios. It has recently entered heavily into sensitive fields like legal, military, and medical systems. The crypto methods designed for encrypting images have been limited and not very suitable for today's usages[5]. For example, the AES system 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 main operations to be tailored and organized for encrypting images [63].

There are two techniques to accomplish encryption: transposition, and substitution [80]. Firstly, the transposition method, which deals with pixels of the original image to map the block of image into other position and shuffle the position of the pixels. Secondly, the substitution method, which is used to change all pixel values [36]. Both techniques achieve confusion and diffusion goals that indicate the principle of cryptography[22].

Image encryption techniques are classified into three techniques; the first technique is permuting pixels. This technique changes the position of pixels which means rearrange the pixels for obtaining the reorganized image. The second technique is substituting values, the value of pixels is replacing according to a key generator. The third technique is a combination of permutation and substitution techniques [58].

Last few years, researchers have studied different problems in image encryption. They have suggested several efficient schemes based on various theories such as; Jacobian elliptic maps [17], Arnold's Cat Map [54], zigzag transformation [57], Baker map [44], discretized tent map[91], Deoxyribonucleic Acid (DNA)[83], S-box transformation [35], Rubik's Cube [88], and the affine transformation [2], amongst many other studies.

This paper is dealing with a symmetric encryption [12], which the process is using the same key for encryption and decryption. It utilizes pseudo random number generator (PRNG) to realize encryption [14]. For example, the studies, in which the same method of image encrypted using PRNG with a different approach is a clear in [14, 38, 60, 71]. PRNG is a procedure of generating a sequence of random numbers with two essential properties; unpredictability and randomness [74].

The term 'pseudo' refers to that the computation numbers are not truly random, however, it's random enough in terms of cryptography application [21]. The pseudo random number

generator requires the initial value called as the seed value [37]. Usually, the user has to provide the seed, then the generator can generate a sequence of random numbers [77]. To ensure a good chosen PRNG for cryptographic purposes, it should pass the United States NIST [15]. It can be conducted on any random generator to determine randomness quality. Poor random generator selection leads to a weak system used in the encryption purpose [21]. For instance, linear congruential random number generator can be expressed in term of linear equation, and can be described as modular linear equations [11]. It can be predicted by guessing internal constants, and detect the state of generation [11]. Due to that reason several scientists are working to improve PRNG weaknesses, particularly the applications of PRNGs used in image encryption [23].

The PRNG can generate long series of random number with good random behavior. For each PRNG should have several properties to consider the right choice. There are respectively, large period, reproducibility, uniformity, consistency, independence, portability, disjoint subsequences, permutations, and efficiency [18]. Therefore, this pseudo random number generators are chosen to be our objective study for encrypting images. We will explain the suitable and effectiveness generator for using in encryption images.

Several researchers are concerned to improve image encryption. The mathematical properties of PRNGs have been widely studied, i.e. researchers developed and proposed a new PRNG in [25, 53, 68, 82, 90]. The practical assistance for determining the strength of RNG algorithm can be providing a strong encryption system. In contrast, we can find that weak randomness is ensuring the scheme of random number generator insecure performance. Due to this security requirement, this research reveals the performance for each individual six of PRNGs using in encrypting images. We will try to convey information about good results for encrypting images based on six PRNGs. The security analysis exposes the highest and worst effective image encryption for many different types of PRNG. So, the research can demonstrate the efficient generators with good characteristics tend to be encrypted images secure totally.

In this paper, we introduce exploring different pseudo-random number generator options for possible improving the security strategy of the image encryption [71]. Our research tested its experimentations on RGB images at the block of pixels levels to obtain cipher image. The work explores the image encryption involving different PRNG within both permutation and substitution sequential techniques, for fair comparisons, i.e. similar in principle to the two steps security presented in [71]. To be specific, the research's main contribution implements the cryptographic approaches via creating the key streams by six pseudo-random number generators (PRNGs) for selecting the appropriate one. The six PRNGs models involved are: (1) Mersenne Twister(MT), (2) SIMD-oriented Fast Mersenne Twister (dSFMT), (3) Combined Multiple Recursive (MRG), (4) Multiplicative Lagged Fibonacci (MLFG), (5) Shift-register generator summed with linear congruential generator (shr3cong), and (6) Modified subtract with borrow generator (SWB). According to this sequence, the pixel permutation and substitution are performed differently, i.e. 6 times applying every PRNG showing different results. All experimental have been conducted on several statistical measures. The results determine the suitable PRNG to use in image encryption and present the final PRNG that achieved the highest effectiveness to secure the image.

The flow of the paper presentation is as follows. Section 2 covers the literature review. Section 3 describes the methodology of our proposed image encryption mechanism using the PRNG algorithm. Section 4 discusses the results and shows the

experimental notes. Section 5 presents the comparisons of our proposal vs. others. Section 6 includes the conclusion of the paper.

2 Literature review

2.1 Related image cryptography

In this section, we will cover the fundamentals of image encryption with two encryption processes as standardization used within this study [71]. This depends on the encryption of images by transposition cipher followed by substitution cipher using random number generators. Coverage will include a brief description of PRNG, and permutation and substitution methods in the context of image encryption and illustrations and highlight the implementation of each operation separately.

2.1.1 An overview of the basic image encryption processes

Image encryption is the process of encrypting an original image in an unreadable form that can only be read and be visible by the owner or authorized person. Many methods can be applied to images based on many transposition or substitution to obtain cipher image [38]. The fundamental concept is as shown in Fig. 1, Alice can encrypt the image before sending it by using her key, while Bob receives the encrypted image. He is able to decrypt it using the same key. The powerful algorithm is used between the two parties in encryption, which makes image information random, and makes it difficult for Oscar to know its contents.

The property and structure of the images have a high redundancy in pixel values and their important size. In addition, the 256 scales are repeated in more than one pixel, which leads to a strong correlation among pixels [6]. These characteristics or image's behaviors are not suitable to some of the behaviors of the encryption algorithms, used in both Symmetric key encryption and Asymmetric key encryption types, such as ASE [73], RSA [73], and DES [73] for their useful use in textual data.

The PRNG is used in cryptography to generate secret keys [78]. The PRNG can produce a set of sequences of random numbers involving two properties unpredictability and randomness [74]. The sequence of the PRNG keeps repeating itself according to the specified length of the

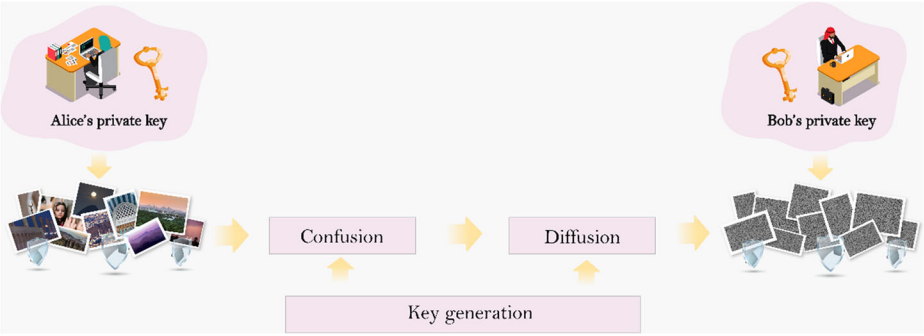


Fig. 1 The general steps for encrypting images

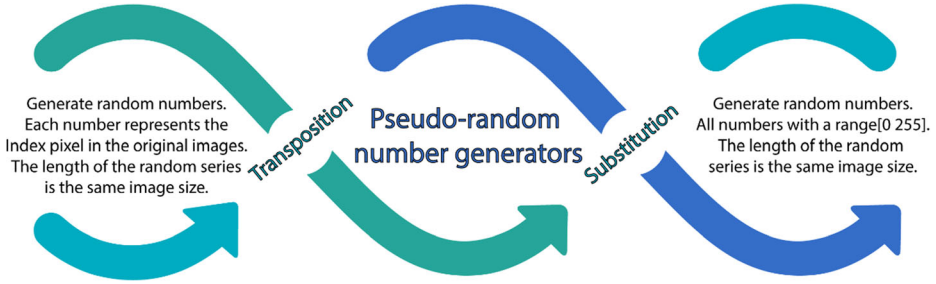


Fig. 2 PRNG for encrypting images

period and produce long sequences of random numbers also unpredictable nature of the produced random numbers.

Figure 2 explains in-depth the technics of PRNGs used in our cryptosystem to investigate their performance. The behaviors of random number generators are unpredictable, randomness, and uniformly distributed. We can generate two keys by using six PRNGs: permutation key and xor key. In the permutation process, we reorder or rearrange the block of the image to shift to another position [40]. This situation remodels pixels of the images and makes them visually random. Furthermore, in the substitution process, change, or replace pixels by substituting the pixel's identity [40]. It leads to replacing pixels with another value by keeping the same position of the pixels. We will introduce in detail manner the two techniques that we will use in image encryption and describe the performance of PRNGs in the two processes.

The basic block permutation for encrypting image The technique of transposition is accomplished by performing some sort of permutation on the images. The identity of the pixels is not altered, also the technique doesn't replace one pixel with another and instead, it switches the location of pixels. The position of pixels in the image is arranged according to the random removal of the association between pixels [19]. The image content would be totally disordered, and it will not remain understandable. Flipping and altering the pixel values orderly can be termed as a confusion concept [89]. The aim of applying permutation is to reduce the possible correlations between the plain images and the cipher images [28]. And, to stand powerful against common differential attack measures as well as to prevent brute force attacks.

Block permutation is a process for the decomposing image into blocks [49]. These blocks are permuted according to a map with a newly replaced index vector. If the number of blocks is very small, namely it is 1 block, then the images are not clear after the permutation [49]. In another word, each block contains 1 pixel and is then transformed to another location [13]. The process of permuting the position of each pixel in the image is based on the linear indexing concept. This indexing treats the matrix as if it is a long vector in which the elements start from the top left corner and go down the first column. Then the second column, etc., along with the last element that is positioned in the bottom right corner of the matrix [70].

For example, in the illustration Fig. 7, we can do the block permutation technique by creating a square matrix with 3×3 sizes, which describes the concept of process. We suppose the square matrix size of the image is 3×3 to maximize the visibility of operations in a simplified way. In addition, illustrating the matrix with an original index position for all elements, P(i,j) is the original position of the block image.

$$\text{Suppose a square matrix, P(i, j)} = \begin{bmatrix} 199 & 140 & 55 \\ 30 & 44 & 71 \\ 18 & 6 & 22 \end{bmatrix} \rightarrow \begin{bmatrix} P1 & P4 & P7 \\ P2 & P5 & P8 \\ P3 & P6 & P9 \end{bmatrix}$$

In this fundamental scheme, the encryption, and decryption process deal with PRNGs to create the permutation key. They produce the integer sequence of values. The generated key is based on a seed value. Entering the same seed is to output the same sequence of values by PRNG. A series of random integer numbers will be generated along with the size of the images, or rather the size of the index length of the matrix, which is 9 blocks long of images, or 9 IDs of the linear vector. It's permuting index length, as it's clear in Fig. 4 the permutation key is [1 6 5 2 4 7 9 8 3].

On the sender's side, the essential matrix can be encrypted using a series of random numbers as a permutation key. These steps are preparation for the encryption process, which uses the PRNG key to scramble and permutes the matrix indexes. The matrix divides into a random number of blocks. Each matrix index, should rely on the key by linking each random number in the cell array to the index. All original index values from 1 to 9 are rearranged into the new array by, shuffling value using the new random indexes obtained from the PRNG. For more clarification, Fig. 3 shows the index of original matrix 1 with value 199 goes to the replaced index number 1 with the same value 199 in a new array, and the index of original matrix 6 with a value 6 goes to the replaced index number 2 with the same value 30, and so on. The rest of the new array indexes are filled in the same method as mentioned before. p'(i,j) refers to block image after shuffling. It's a swapping position, by mapping key permutation with P(i,j) to result in the scrambled matrix.

$$\text{Cipher matrix, C(i, j)} = \begin{bmatrix} 199 & 30 & 22 \\ 6 & 140 & 71 \\ 44 & 55 & 18 \end{bmatrix} \rightarrow \text{p}'(\text{i, j}) = \begin{bmatrix} P1 & P2 & P9 \\ P6 & P4 & P8 \\ P5 & P7 & P3 \end{bmatrix}$$

The output on the sender side from this technique is the scrambling matrix. The writing result from this technique output is, to ensure securing the information and keep it unnoticed, or visible. After the data transfer to another side, it is read as a scrambled matrix. Then the technique produces the same series of random numbers, which the sender used before. Each random number refers to the original index of a matrix, all block IDs go to restore, and order according to the obtained series random numbers. Index 1 in the scrambled matrix, linked to

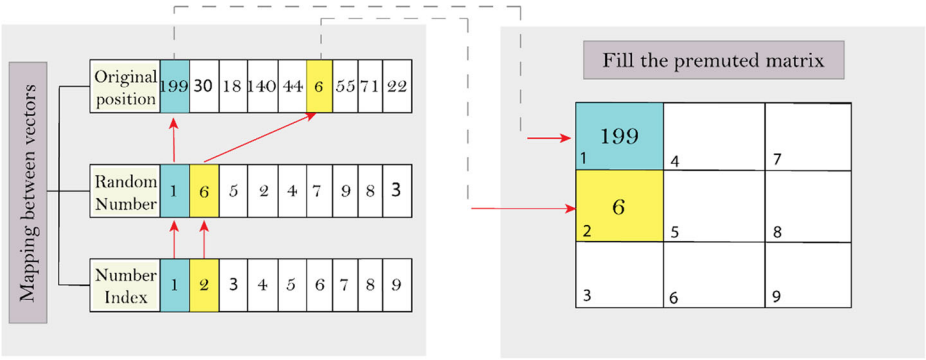


Fig. 3 General illustration for scrambling matrix

random number 1, and index 2 in the scrambled matrix linked to next random number 6 to reorder the original matrix and so on along the scrambled matrix until the whole data is decrypted.

The research studies six of PRNGs for encrypting RGB images. All six random generators are underperformance analytic procedures. For each generator, we study their efficient use to encrypt the image. Each of the random series is used as a new indicator of pixel position and similar in principle to the work presented in [71]. The original images after the change, have the indexes of random series. The performance of the six generators is studied in terms of applying image encryption to all six generators and recording several security measurements.

The length of the elements for each vector of the different PRNGs is from 1 to the length of the selected image. The behaviors of all key generations are nonrepeating random integers. As shown in Fig. 5, we selected the color Lena 256 × 256 and MT generator. The random permutation key is 1-dimensional with a length of 1 × 65536. It takes the same length of the original image, as the color Lena 256 × 256. The frequency of each 65,536 samples of random integer permutation is 1. This confirms that each number in the vector appeared one time. It is a permutation of a set (1, 2, … 65,536). Each vector content depends on the value of the seeds and the type of algorithms used in the permutation process. The distribution of each number of all series are integer, unique, and unsigned.

The basic pixels substitution for encrypting image One of several systems used to encrypt images and diffusion is a substitution cipher. This kind of cipher, is a method of replacing bits

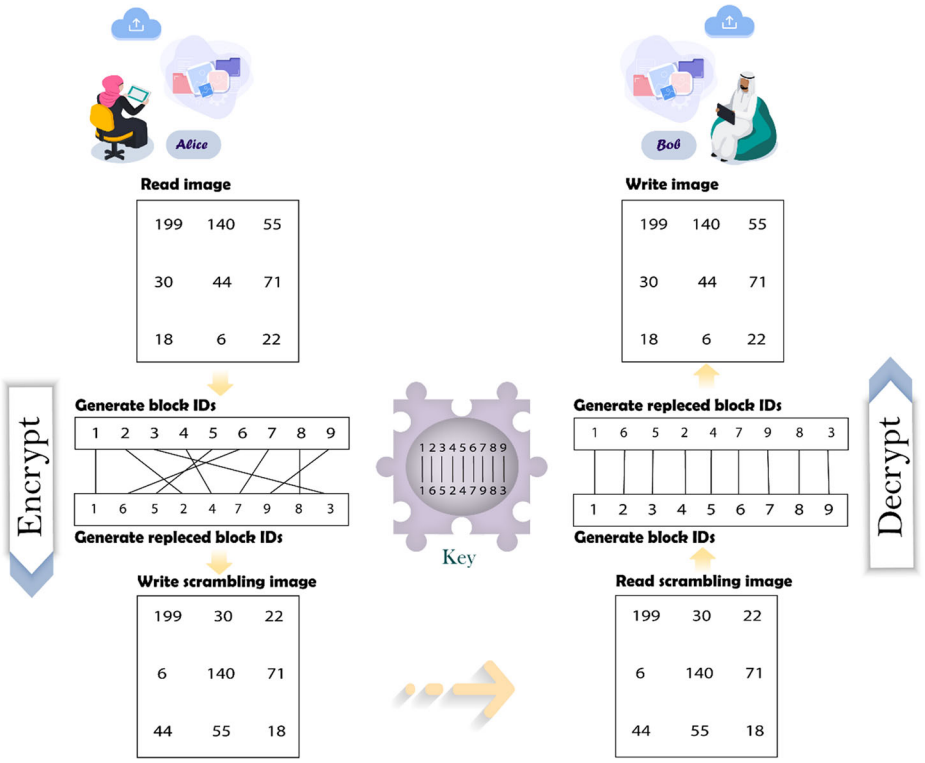


Fig. 4 Permutation blocks image and de-permutation process

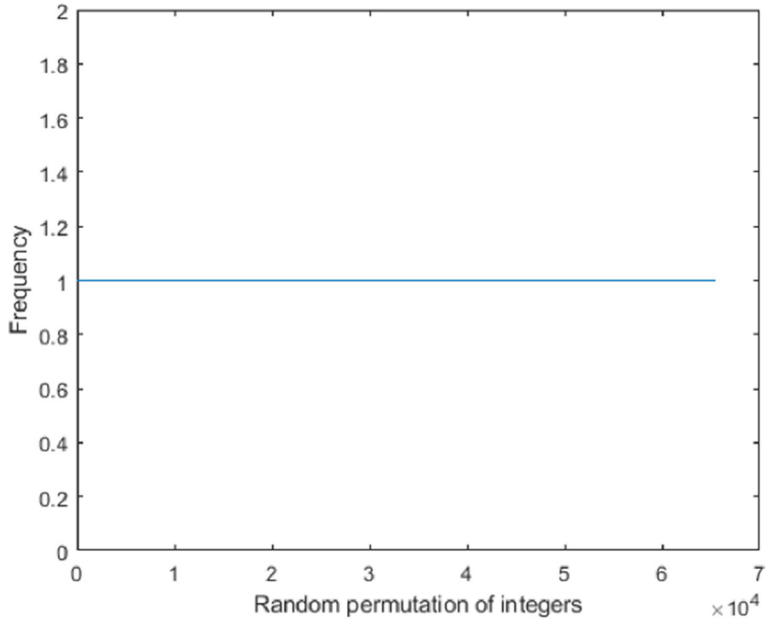


Fig. 5 MT generator frequency

with substitution way using a particular algorithm. The algorithm determines how the replacement occurs, and it is based on a key. Therefore, the receiver of the encrypted message, must know the algorithm used for encryption, and key mechanism to apply the decryption. When the receiver receives the encrypted image, the receiver will use the known substitution algorithm between the two parties to decrypt the images and show the original image that the sender wanted to send.

When making substitution and transposition ciphers in images, there are obvious differences between the two methods. In the process of transposition, the primary target is confused changing positions without being affected on pixel values. Otherwise, substitution cipher maintains the same sequence of pixel positions and modifies the same pixels with other values determined by the algorithm and the key used in encryption.

Substitution cipher is utilized pseudo-random numbers to generate keys that is a suitable structure, for encrypting the images. Each bit of the image is linked with the entire key by XOR operation, to replace the plain image bits in the encrypted image bits. Any changing on bits of the key, will alter the cipher image entirely. The XOR operation appears in a binary sequence with two values 1 or 0 and the probability of 0 appearance occurs when the two entrances are the same while the probability of 1 value appearance occurs when the two entrances are different.

For instance, if we assume the 3*3 matrix refers to a plain cropped image, we can estimate the general processes and explain them in detail as following: To cipher color image in a range of [0255], we use XOR operation by generating key in the same image class. Anyway, the P refers to the plain image with 3*3 sizes and K_E refers to the key of encryption with the same size of P. The K_E is equal to K_D that is used in an inverse way to decrypt the image. Each pixel shall XORed with the corresponding key. The results of XOR operation are shown in the following Table 1.

Table 1 A simple illustration example of the xor process that replaces each pixel with the corresponding value of the key

Plain image	XOR	Key	Result
199	⊕	25	222
30	⊕	56	38
18	⊕	238	252
140	⊕	84	216
44	⊕	52	24
6	⊕	43	45
55	⊕	129	182
71	⊕	185	254
22	⊕	146	132

$$\text{Suppose plain image, } P = \begin{bmatrix} 199 & 140 & 55 \\ 30 & 44 & 71 \\ 18 & 6 & 22 \end{bmatrix}$$

Generating key by using Mersenne Twister generator.

$$K_E = \begin{bmatrix} 25 & 84 & 129 \\ 56 & 52 & 185 \\ 238 & 43 & 146 \end{bmatrix}$$

Then cipher image E: $C = P \oplus K_E$

$$C = \begin{bmatrix} 222 & 216 & 182 \\ 38 & 24 & 254 \\ 252 & 45 & 132 \end{bmatrix}$$

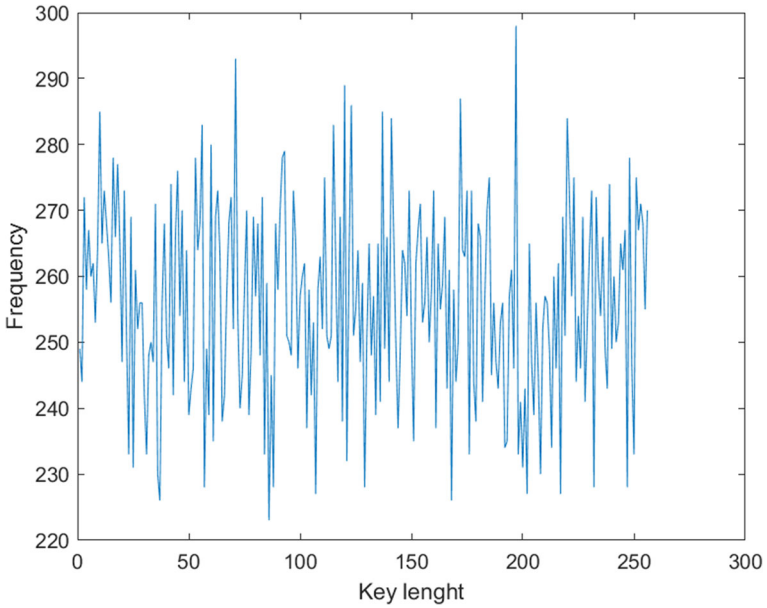


Fig. 6 XORkey frequency by using MT generator

The real example of performance analysis on samples of XORkey is as shown in Fig. 6. We create a key using the Mersenne Twister generator for the same size as 256 for Lena's color image. The random number series are integer numbers. Key behavior appears to have a low and maximum value in the period [0255], and has recurring numbers so when the series reaches the specified period it produces the same period again until the key size is completed.

The combination of the transposition and substitution processes make the security system stronger and more complex. The sequence of the two processes depends initially on the image encryption by the transposition cipher and secondly on applying the substitution process to the result of the image we obtained from the transposition cipher. So, one of our objectives in this study is to study the performance of all six PRNGs to generate keys that will be used to encrypt the images and conduct tests on the results of the encrypted images.

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: Ciphered image
1. Ensert two inputs: image size, and seed value.
2. Generate x sequence by using MT generator.
% Divide the 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 the image with index x sequence.
14.End for.
15. Convert cell2mat and save the image as a 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.

This research aims to recommend encrypting images in a suitable way, and determine the efficient PRNGs between usage different options of PRNGs. The study records the several parameters to determine the effective security way for encrypting images by three techniques of encryption by using six of PRNGs. All three techniques of experiments (permutation, substitution, and combination two operations) are shown in Fig. 7. The three experiments should show differences in results. Each experiment should analysis with several

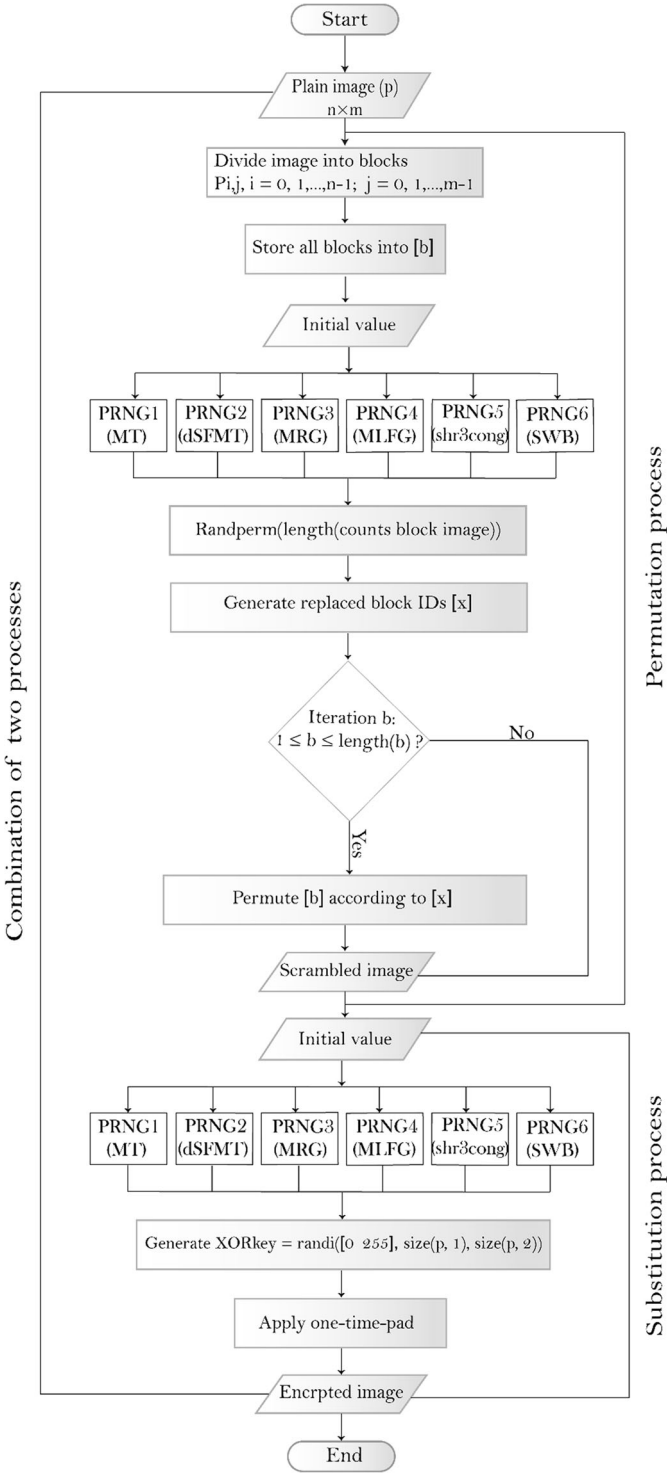


Fig. 7 Image encryption using transposition and substitution operations and PRNGs

measurements and by comparing the results, we can record the high score in results. The high results can be a good effect on cost-effectiveness. Otherwise, when the results record low score in measurements, the cost- effectiveness refers to the worst way of image encryption.

2.2 Related random number generator (RNG)

Random number generators (RNGs) are essential bases for cryptography [39]. For example, they are commonly used as trustworthy key generators for public-key cryptosystems e.g. RSA, symmetric key e.g. stream cipher, and as sources of passwords. Random numbers are intrinsic to the account in certain algorithms such as DSA or protocols such as zero-knowledge proof [27]. RNGs are a principal method used for producing a sequence of numbers by using software or hardware types of generator between specific interval [min, max], this sequence has a feature such as non-deterministic in nature.

When the generator computes the values mathematically, each sequence of values should be independent of the others. Some features which are considered to be desirable RNG:

- The RNG should be fast enough in producing a large sequence of random numbers.
- The RNG should be more secure against attackers.

There are two types of random numbers generators (RNGs) founded on vary source of randomness as shown below [50]:

- True random number generators (TRNGs):

TRNG presents genuinely random data obtained from non-deterministic events that exist in nature. It is a non-deterministic generator utilizing a physical process. The entropy of the output of the generator depends upon the entropy of a source. Physical phenomena generally used in the production of random numbers are thermal noise, radioactive decay, and the cosmic microwave background [10].

- Pseudorandom number generators (PRNGs):

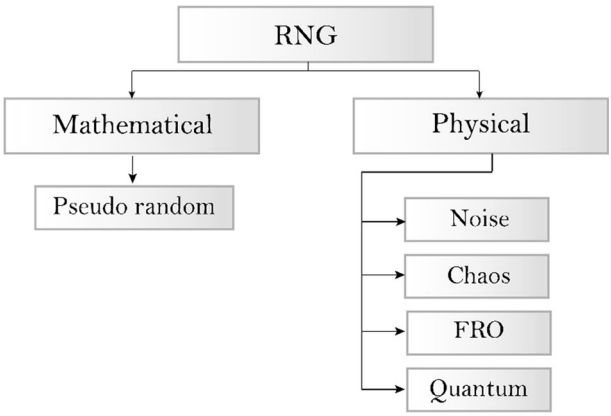


Fig. 8 Two categories of RNGs types

Table 2 Comparison of the three main qualities between PRNG and TRNG [33]

Features	PRNG	TRNG
Determinism	Deterministic	Nondeterministic
Efficiency	Excellent	Poor
Periodicity	Periodic	Aperiodic

PRNG is explained as a mathematical algorithm for generating a sequence of numbers with random features. It is a deterministic generator utilizing a mathematical algorithm. Due to that, it is not truly random, it completely depends on the initial value, called as the state of the PRNG. The initial value needs to be unpredictable and random [62].

The classification of RNGs is shown in Fig. 8, depending on the form of the algorithm used, mathematical and PRNG can often be classified into many types each type has its own mathematical calculation; and on the other hand, TRNG has many structures based on various entropy sources, it attracted many researchers and research is acquiring an impetus yet, recently, it can be classified into four sources: noise, chaos, free-running oscillator (FRO), and quantum RNGs.

In comparison, the merits of PRNGs are totally different from TRNGs. TRNGs are considered roughly ineffectual compared to PRNGs, it extremely spends a long time to generate numbers. They are too non-deterministic, i.e. it is not possible to repeat presented

Fig. 9 Most suitable generator for each application

sequences of numbers, while the same series can definitely occur many times by chance. TRNGs don't have a specific period [33]. The two types of RNG can be described by three main characterizations as it is shown in Table 2.

Based on different characterizations between PRNG and TRNG, the appropriate RNG type with high characterization is suitable approximately for the set of applications. For example, using TRNG for games and data encryption is suitable. Whereas excellent efficiency and deterministic nature of PRNGs make them more appropriate for simulation and modeling applications. The following Fig. 9 presents which applications are better supported by its generator type [33].

In this research, we compare six types of PRNGs to determine their suitability for image encryption. All PRNGs involved in this research appear as follows:

A. Mersenne Twister generator (MT) [48]. (It is known as *mt19937ar*)
B. SIMD-oriented Fast Mersenne Twister (dSFMT) [64]. (It is known as *dsfmt19937*)
C. Combined multiple recursive (MRG) [41]. (It is known as *mrg32k3a*)
D. Multiplicative Lagged Fibonacci (MLFG) [47]. (It is known as *mlfg6331_64*)
E. Shift-register generator summed with linear congruential generator (shr3cong) [45]. (It is known as *shr3cong*)
F. Modified subtract with borrow generator (SWB) [46]. (It is known as *swb2712*)

2.3 Considered image encryption work

The researchers in the digital security field have proposed many digital encryption methodologies that are specialized 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.

A. *Image encryption utilizing scrambling*

From the practical point of view, the scrambling of the image in classical cryptography is analogous to the substitution process. In modernistic cryptosystems, the scrambling of the image is also utilized in pre-processing and post-processing for encrypting images. This also plays a significant role in cryptography. Experts have recently carried out extensive and in-depth analysis work which has accomplished fruitful results. Arnold transform, also known as cat map, which Arnold proposed while he was researching Ergodic theory. Arnold transform was suggested for the transformation of the large dimensions [20]. Furthermore, researchers suggested other image scrambling approaches, for example, affine transformation, the MagicCube, and Baker's transformation. The scope of several advanced types of research involves various methods of scrambling image as follows:

Sarma et al. in [67] propose image encryption methodology based on scrambling techniques to change pixels positions. The methodology is based on two keys, which are used as input to their proposed algorithm to produce a sequence of random numbers that will be used to scramble the image to be scrambled. 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 scrambled image. The approach uses two keys as input instead of incorporating a password. The scenario is inappropriate to use real numbers.

T. Sivakumar and K [71]. Gayathri Devi proposed irregular stage of squares and performing XOR activity over the permuted picture with the key produced by utilizing Lagged Fibonacci Generator (LFG) for image encryption. The graph square shows the progression of procedure for there's proposed technique. These methods are our enthusiasm to contemplate the viability encryption way and research various forms of PRNGs.

In 2018, a novel PRNG based image encryption methodology was proposed by S. Saha et al. [63]. 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-level image encryption, where in the first level the shuffling of the pixels is based on the PRNG algorithm LFSR. The second level, encryption is performed where the XOR is applied on the pixels to substitute their values by replicating the rows with columns to produce the final encrypted image.

Ramasamy et al. [59] proposed a novel symmetric key generation to encrypt color image. They use block scrambling, modified zigzag transformation, and enhanced logistic-tent map. Block scrambling is conducted on color image, and the total number of blocks is 64 blocks. Then, a modified zigzag transformation is applied on the obtained result from block scrambling technique. So that the key is generated by using enhanced logistic-tent map and using secret key to XOR RGB image after performed zigzag transformation.

Zhou et al. [92] implemented a quantum image algorithm focused on Arnold transform created to encrypt location information, whereas the gray-level information encrypted through the double random-phase processes, which is heuristic to present more techniques of image processing into quantum image encryption. The proposed algorithm for image encryption has reduced computational complexity than its traditional predecessors.

B. Image encryption based on entropy concept

In 1948, the inventor of the theory of information developed the principle of entropy-based information in thermodynamics [69]. Thermodynamics entropy explains the disordered state of the physical system, and thermodynamics entropy is considered as a measure of the degree of disorder. Similarly, we also take into consideration information entropy to be a measure of the degree of signal disturbance that is used to describe information instability. When the receiver obtains information, the ambiguity is reduced, the entropy of the source is lowered to acquire the information. Image encryption aims to render the encrypted image in a state of disorganization. Consequently, image encryption is intended to improve the original image's security. Pixel replacement and pixel diffusion are both successful ways to change image encryption pixel values which can reduce pixel correlation and increase the entropy of information. The value of each pixel will be changed individually in the pixel substitution process which is not connected to others. Taking into consideration that image encryption needs high speed, and the image substitution normally utilizes "Exclusive OR" operation [76]. Lately, this method is involved in several types of research, we can demonstrate them bellow:

Banthia and Tiwari [14], show a suggestion for encrypting images, by using two techniques of random generators. The two random generators are linear congruential generator, and Logisitic map. The linear congruential generator permutes the position of pixels. Rows and columns reorder according to number generated. The Random number is generated by Logisitic map. These numbers are used to reorder the position of rows, columns, and pixels.

A. Ramesh and A. Jain [60] have proposed a new hybrid image encryption methodology using Pseudorandom number generators. Their methodology combines 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 the key, and the resulting sequence of random numbers is used to swap the position of each pixel of the encrypted image with another pixel.

Another algorithm was proposed in 2015 by V. Kapur et al. [38] to encrypt and secure images using Pseudorandom number generators. Their proposed methodology is a two-level image encryption algorithm, where the first phase uses the Linear Feedback Shift Register algorithm to swap the rows and swap the columns of the image 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 produce the final ciphered image.

One of these plans examined by Imam Saputra [66] contemplated randomization of images utilizing just one occasion cushion calculation which is an exceptionally amazing calculation and very hard to unravel by cryptanalysis within the event that it's an extended and arbitrary key. Utilizing a produced key utilizing an instantaneous compatible arbitrary generator will create an irregular key.

C. *Image encryption based on Chaotic scheme*

The chaos theory is a branch of mathematics that focuses on the actions of the Dynamic structures extremely sensitive to initial conditions. The chaos theory implies the nonlinear dynamic system's random phenomena; this phenomenon is non-convergent, non-periodic and is highly susceptible to initial and external parameters. Chaotic systems are nonlinear dynamic systems [42] with strong randomness that can generate pseudorandom series and is useful for data encrypting. In 1997, the chaotic systems were introduced for data encryption, and Fridrich introduced the image encryption method at first based on a chaotic system [26]. Recently numerous studies discussed this approach as follows:

Linda and Karim[29], proposed a combination of pseudo number generator to generate an encryption key. The key generated is used to encrypt an image in two stages. The two stages are confusing and diffusion. Pixel permutes and changes according to the combination keys of two Logistic map.

Rohith S. et al. [61], the authors suggested a composite method to generate the encryption key used, for encrypting the gray image. The method depends on the image encryption by using the theory of chaos. The two random generators Logisitic map, and LFSR are used to generate two keys, and apply the XOR process to obtain the final key that shall be used to encrypt the image. This approach is better in security because it depends on the combination of chaos systems to provide the key instead of generating key by single chaos.

Chanil and Lilian [52] have introduced a new color image encryption with a better new one-dimension chaotic system. This approach is better chaotic performances than the old one-dimensional chaotic map. The method produces chaotic sequence for shuffling pixels positions and obtains scrambled images. After that, diffusion operation is applied to a permuted image.

Several image encryption algorithms have been evolving in recent years to increase the capability of transmission and encryption such as in Ye et al. (2020) [84], optical multiple-

image encryption systems. Ye et al. suggested an encryption technique for multi-image founded on quaternion discrete fractional Hartley transform (QD-FrHT) and enhanced pixel adaptive diffusion. Moreover, in this method, the plain images are compressed into four fusion images by using two operations; discrete cosine transform (DCT), and Zigzag operations. Then the four resulting images are defined as the quaternion algebra.

Another study suggested image encryption using the phase-truncated short-time fractional Fourier transform, along with the hyper-chaotic system [87]. PTSTFRFT is distinguished from standard phase truncation coding in that it has been paired with wave-based permutation. This is achieved according to create the encryption unite to code divided image. In this encoding, the confusing phase information is recombined with the amplitude information.

3 Proposed method

This paper presents different keys of six types of PRNGs based on color image encryption. The research examines different possible techniques for encrypting images. Depending on this research, we can introduce an effective generator for using to encrypt images. The system checks the performance of individual PRNG by using three techniques for encrypting images, i.e. we use MT generator in different phases for permuting image technique, substituting image, and combined of two techniques. Then, we work on estimating the performance for MT generator at each state of three techniques. The main processes consist of encryption and decryption processes.

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 as well as the secrete pin-code that 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 output the encrypted image.

On the other hand, at the receiver side, the proposed methodology works in the reversed order. 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, the steps were taken previously for encrypting the image is now taking in the revised order to decrypt the image and outputs the decrypted image to the receiver user.

The main objective of this research is to determine and presenting a secure way to encrypt images based on different types of PRNGs. The implementation of encryption images is to use the same basic scheme of encryption and decryption with all 6 PRNGs. This work shows the performance of different types of PRNGs. Each random generator individually can produce a high or slow score rate in several measurements. The system checks the good and worst results of image encryption based on PRNGs with three methods of experiments of encryption.

The research examines three outputs images: (a) a permuted image using a permutation blocks process, (b) a ciphered image using XOR, and (c) a permuted image encrypted using XOR. Several statistical parameters of the three images are computing to demonstrate strong performance between methods and compared with each other. The main three methods experiences for encrypting images are discussed as follows:

A. Approach 1: Permute images by using only transposition cipher/permutation. The method is done with the same existing study in [67, 71]. The user should insert the initial value. It appeared in Fig. 10. Each pixel was reordered randomly. Pixels positions in original

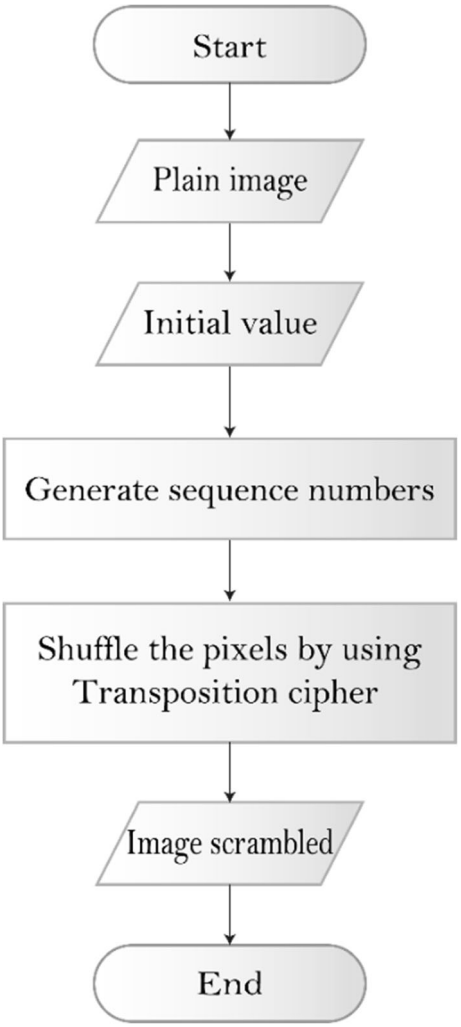


Fig. 10 The first experiment

image change to new positions. This method leads us to perform analysis on scrambled images and study the performance for all 6 permutation keys of PRNGs.

B. Approach 2: Substitute each pixel with a new value by using a substitution cipher. Similar work is in a one-time pad algorithm [66, 71] with different PRNGs. In Fig. 11, we apply only XOR operation to encrypt images. The research records the differences with previous method 1. The outputs from this method are placed in several measurements. Each usage of 6 XOR keys of PRNGs indicates different results. All keys have been analyzed individually by using the National Institute of Standards and Technology of the U.S. Government (NIST).

C. Approach 3: Combine two operations to encrypt images, respectively as shown in Fig. 12. In this method firstly, use the generated sequences to randomly shuffle the pixels.

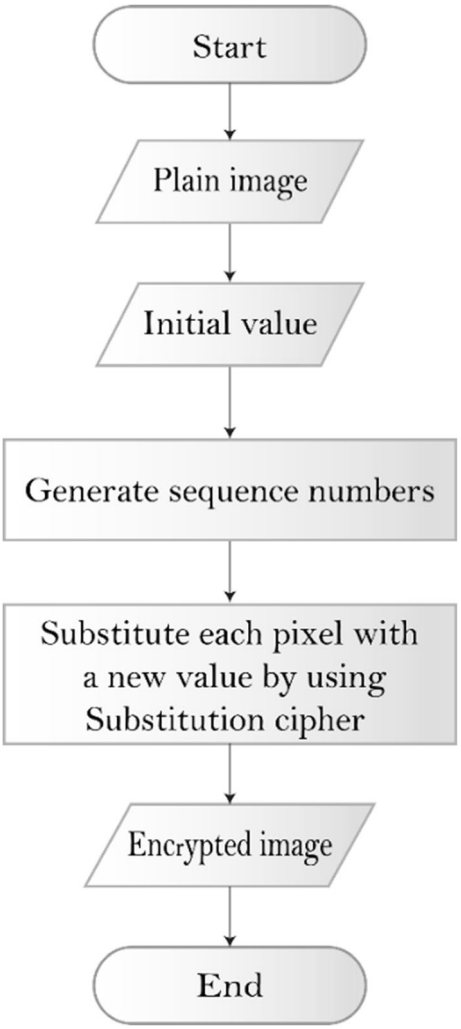


Fig. 11 The second experiment

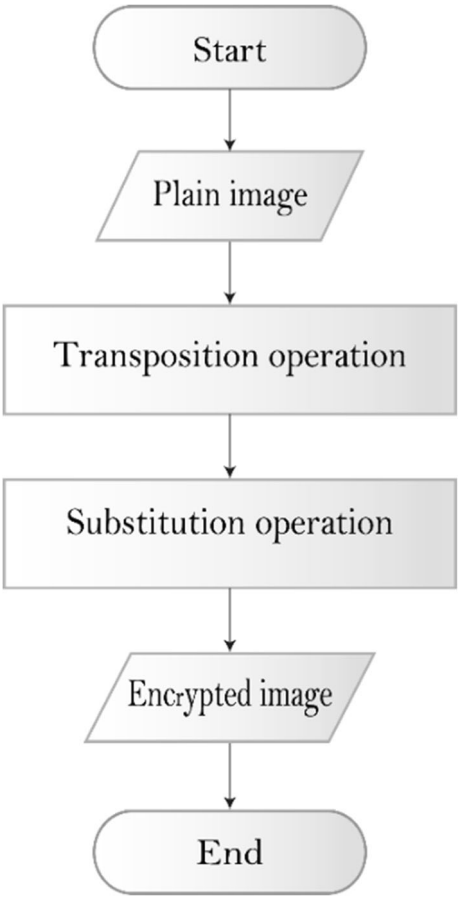
Secondly, apply substitution, for XORing each pixel with a key. Thirdly, the encrypted image is evaluated and compared with other previous methods.

GUI implementation for encrypting RGB image through single PRNG The system implementation for encrypting RGB image of a single type of six PRNGs is shown in Fig. 13.

The window that appeared after the implementation contains two panels:

A. The left panel specializes in encrypting an RGB image by splitting three channels of the color image. It contains a button to select a color image that will be encrypted.

Fig. 12 The third experiment



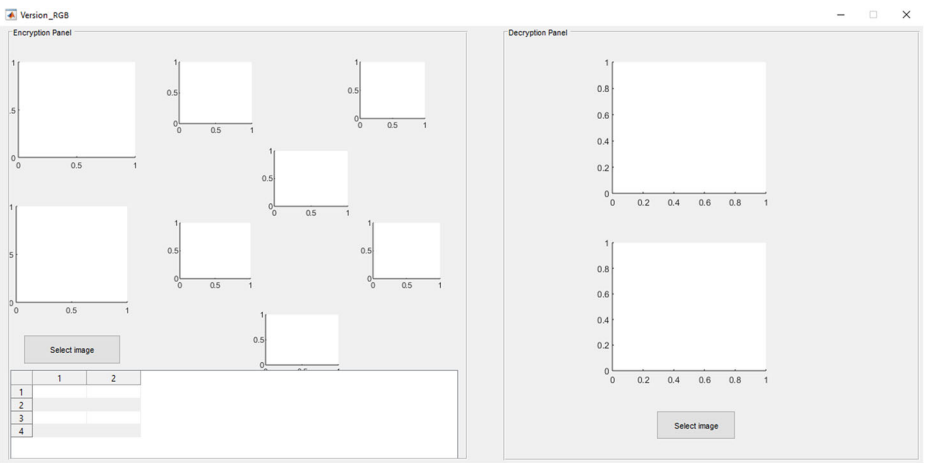


Fig. 13 The graphical user interface window

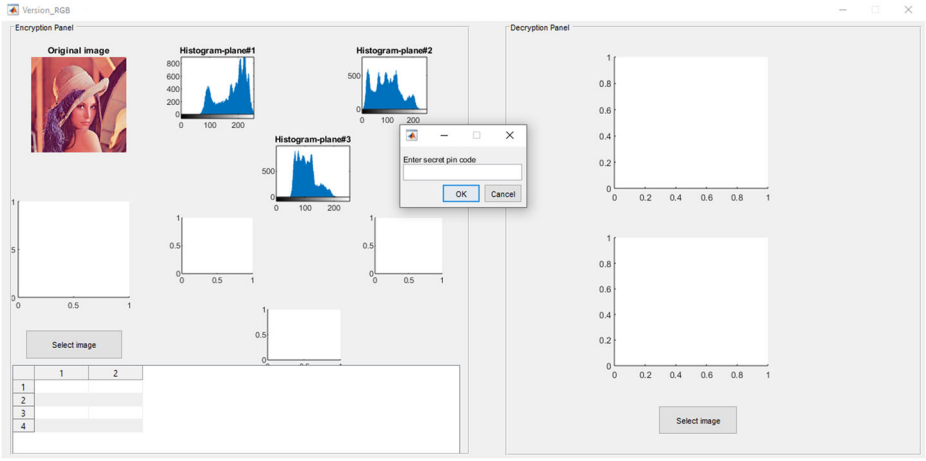


Fig. 14 The system gets the inputs for preparing color image encryption

B. The right panel represents the receiver side. It takes a reverse operation for encrypting color images. The decryption side contains a button to select an encrypted image that will be decrypted.

When, a color Lena image uploads and runs the process for encrypting Lena image by using MT generator, the window can be shown as in Fig. 14. The requirement to continue the encryption process is to insert a PIN. Besides, each time the user enters a different pin code, a different sequence of random numbers will be produced accordingly, and hence the output of the encrypted image will be different and unique from any previous encryption processes. 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 final output from the encryption process is shown in Fig. 15. The performance metrics are applying for each plane; R plane, G plane, B plane, and for combining RGB.

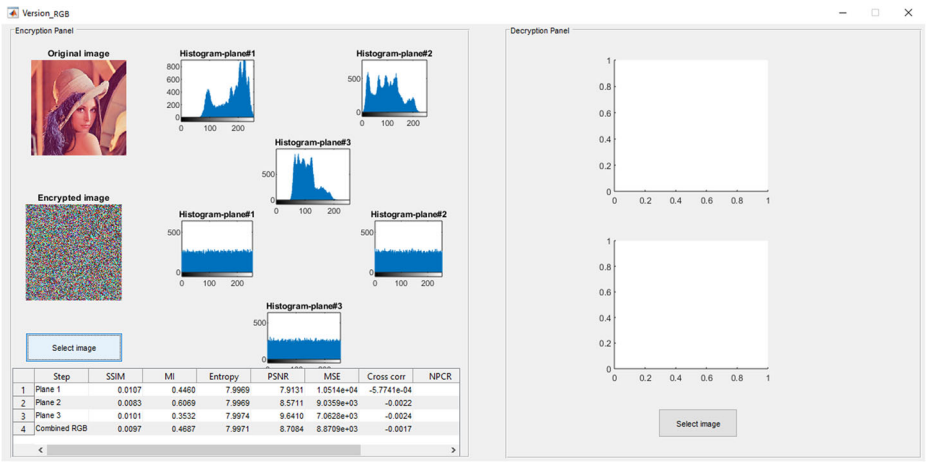


Fig. 15 The system obtained the final encrypted image and results of performance analysis

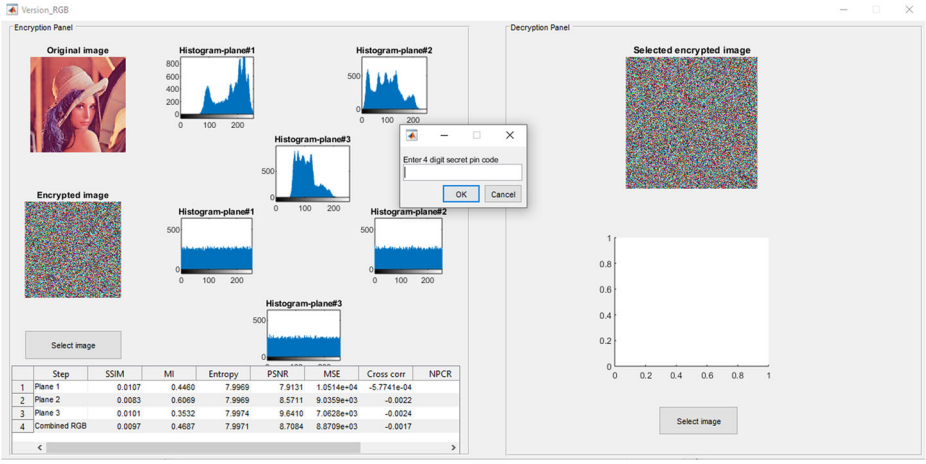


Fig. 16 The system gets the inputs for preparing color image decryption

The right panel is a decryption process. The system gets two inputs from the user to decrypt; the encrypted image that the user wants to decrypt, and PIN, as shown in Fig. 16. The seed is the produced sequence of numbers. It is produced to encrypt the image which can be reproduced by the receiving user to decrypt and hence receive the encrypted image.

The final decrypted color image is done by reversing the operation of encryption, as shown in Fig. 17. If the user entered the pin code incorrectly, he/she could not read the image and decrypt it. That user will only be able to decrypt it who knows the secret pin code that the sender used to generate the image. When the correct pin code is provided, then the same shuffled and substituted sequence will later be obtained.

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

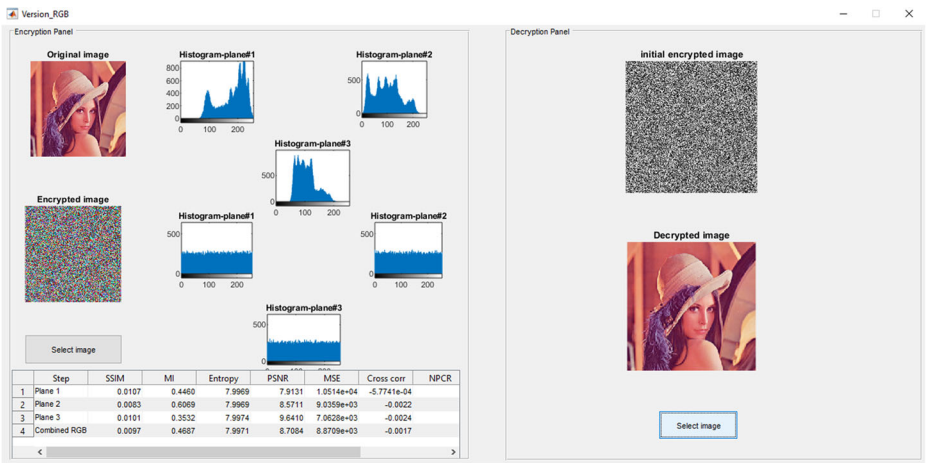


Fig. 17 The system obtained the final decrypted image

3.1 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 [63]. The cryptography algorithm is tested on RGB images.

3.1.1 First operation: Transposition/ permutation block

Step 1: Read the plain color image from the user; RGB channels image.
Step 2: Get the Secret personal identification number (PIN) code from the user, as a seed of PRNG.
Step3: Divided 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 with the same pin code (seed) to generate a random sequence. The six PRNGs that are considered in this study: (1) Mersenne Twister, (2) SIMD-oriented Fast Mersenne Twister, (3) Combined Multiple Recursive, (4) Multiplicative Lagged Fibonacci, (5) Shift-register generator summed with a linear congruential generator, and (6) Modified subtract with borrow generator.
Step 5: Shuffle the positions of the pixels according to the PRNG results (generated random numbers). The pixel permutation is performed by all six selected generators in this study so that each image will be scrambled once time at each time we select one generator from the six random generators.
Step 6: Display the scrambled image after transposition and examine the security of the results.

3.1.2 Second operation: Substitution / XORing pixels

Step 1: Use the same generator of PRNG for every transposition operation. The range of each key from the six PRNG is (0 to 255). The key size is the same image size 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 corresponding key value of the pixel value.
Step 3: Display the encrypted image after substitution and examine the results.

3.2 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 that was used for the same image encryption. To be specific, the decryption process can be outlined as below.

3.2.1 First decryption operation: XOR (de-substitution)

Step 1: Read the encrypted image from user and get the length, breadth, and num of channels.
Step 2: 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.

3.2.2 Second decryption operation: Pixel unscrambling (de-transposition)

Step 1: Use the same pin code with the same PRNG algorithms to generate the same six PRNG 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.

4 Results and discussion

This section studies the results and statistical comparison remarks applying the approach images cryptography. The test is noted in every stage of encryption to build a fair exploration. The study tested its philosophy on one form of RGB images. Our security analysis is made by studying several metrics showing results of key sensitivity, visual testing, histogram, Chi-squared, Mean, median, variance, standard deviation, entropy, correlation, MSE (Mean-Squared Error), PSNR (Peak Signal to Noise Ratio), Signal to Noise Ratio (SNR), Mean absolute error (MAE), Normalized Absolute Error (NAE), structural similarity index (SSIM), Structural dissimilarity (DSSIM), Universal Image Quality Index (UIQI), IQMs based on difference distortion, 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) i5-7200 U running CPU @ 2.50GHz 2.70GHz, 8GB memory, and 64-bit Operating System, ×64-based processer, i.e. 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. All experiments of encryption methods are tested on 5 color images from the USC-SIPI image database [75].

4.1 Visual testing

As shown below in the following Fig. 18 by the visual testing with abstract sight, nothing can be inferred from the encrypted images with a combination of permutation and substitution methods proving that the image encryption method is visually effective. The figures show samples of images in which we validated our algorithms and present each original image with corresponding encrypted images of the steps of the operation. The study presents encryption images by permutation images method to prove its experimentations. There is no change in the

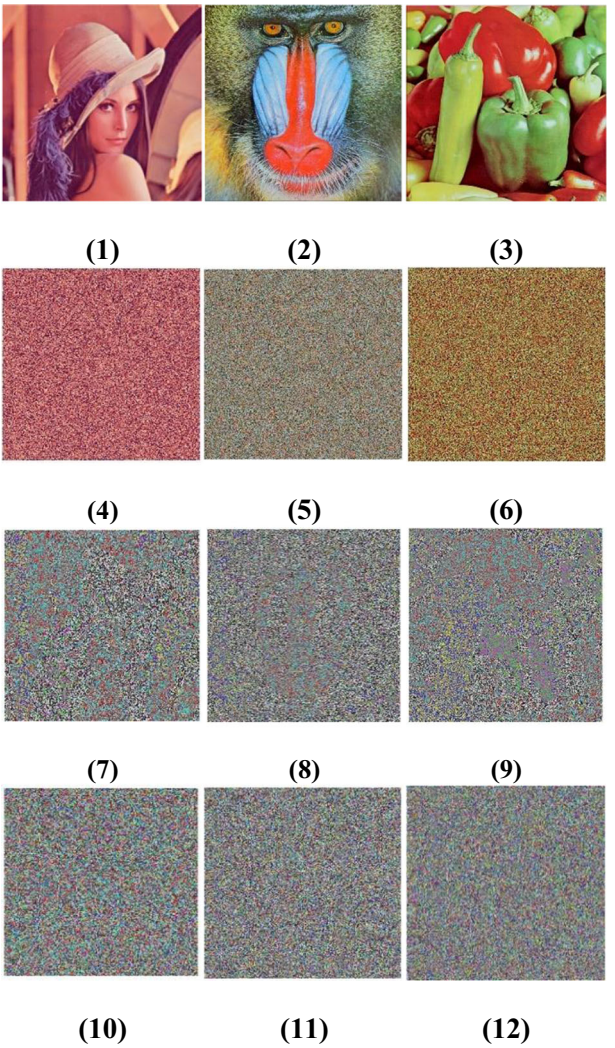


Fig. 18 Image encryption by using MT generator. (1–3) is an original image. (4–6) is scrambled images by using the permutation method. (7–9) is ciphered images by using the substitution method. (10–12) is an encrypted image by using a combination of two methods of permutation and substitution

pixels value. While using the substitution method, the results show nothing in the correlation between pixels except the change in the intensity of the pixels.

4.1.1 Histogram

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 there in each one of the 255 grey levels. It is 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

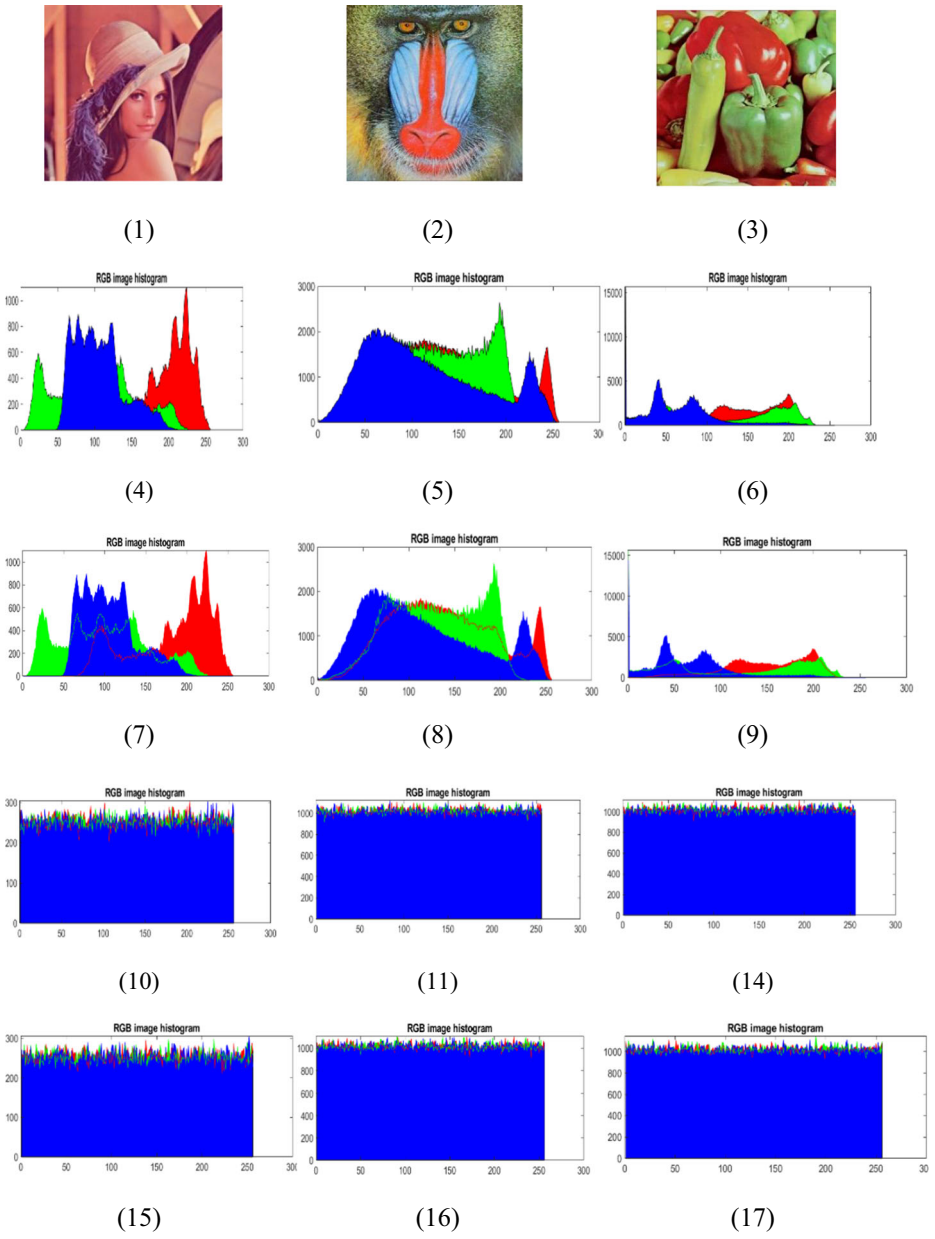


Fig. 19 Histogram of images encryption by using MT generator. (1–3) is original image. (4–10) Histogram of scrambled images by using permutation method. (11–15) Histogram of ciphered images by using substitution method. (16–20) Histogram of encrypted image by using combination of two methods permutation and substitution

number of pixels on the y-axis for each color that is presented on the x-axis. It estimates the distribution of color in encrypted images for image encryption methodology.

We work through our methods 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 color images to view the pixel intensity by graphically displaying the number of pixel distribution of the images after the encryption. Figure 19 are shown differently after implanting each of the cryptographic stages of the study for samples of color images and their cipher images. The cipher methods with the three different outputs can show differences in comparing histograms.

The results are good in using a substitution cipher method, and in using a combination of permutation and substitution ciphers. The bars of the histogram diagram of the images after encryption are evenly distributed among the 255 levels, which indicates that the quality of the images encrypted by the image encryption method is good. The difference in the distribution of pixel positions for all images has been reflected in the pixel shuffled level. Also, the obvious difference in flattening the histogram of the image means a change in the pixel values resulting from XORed images. The histogram of scrambled images by using the permutation method has a clear distribution of pixels in the graph allowing intruders to expect reading the image information. While encrypted images by substitution and combination of two methods distribute pixels in the graph, they are uniform and flat. That means that unauthorized can hardly expect that the image information and the strength of resistance to the statistical attack are enough. The randomness resulting from the distribution of histogram through applying two methods shuffles followed by XORing algorithm ensures good effect and high confidentiality.

4.1.2 Chi-Square test

The uniformity is demonstrated by the chi-square test. It is calculated by the following Eq. 1 [24]:

$$x^2 = \sum_{k=1}^{256} \frac{(O_k - E_k)^2}{E_k} \tag{1}$$

The O_k refers to the observed frequencies of the gray intensity values (0–255). The E_k refers to the expected frequencies of the gray intensity values (0–255) [24]. The performance of an encryption algorithm is better when the value of chi-square χ 2 is low. The lower value means the distribution of the histogram of an encrypted image is uniform [65].

For comparison between three methods of image encryptions by using each of the six PRNGs techniques, the chi-square values of the encrypted images produced for three methods of encryption by using each PRNG separately. Chi-square was compared as shown in Table 3. The min function is computed for chi-square in three methods of encryption for all different types of the six PRNGs. The min value indicates the best value of chi-square among the other three experiments of image encryption. After the observed the best value of chi-square, we computed the med function to analyze the moderate performance among the other three methods of encryption. Otherwise, like max function is computed for chi-square in three methods of encryption for all different types of the six PRNGs. The max function indicates the worst value of chi-square among the other three experiments of image encryption.

The comparison provides the chi-square for experiment 1, which is encrypted Lena, Baboon, Peppers images by permuting the blocks according to each of the six PRNGs. It is shown in Table 3 that chi-square in permutation method for encrypted Lena, Baboon, Peppers images is the highest, this means that the distribution of the pixels is not changed. Furthermore,

Table 3 Chi-square test for original and encrypted images

Original images with X^2_{test}	planes	PRNG No.	PRNG Algo	Encrypted image by three cipher process			Functions		
				Experimental 1 Transposition	Experimental2 Substitution	Experimental 3 Combination of two methods	Max	Med	Min
				X^2_{test}	X^2_{test}	X^2_{test}			
Lena 256×256 (6.5306e+04)	RGB	1	MT	65,305.6	275.25	225.65	65,305.6	275.25	225.65
		2	dSFMT	65,305.6	283.61	245.19	65,305.6	283.61	245.19
		3	MRG	65,305.6	274.03	272	65,305.6	274.03	272
		4	MLFG	65,305.6	261.43	240.22	65,305.6	261.43	240.22
		5	shr3cong	65,305.6	226.83	286.52	65,305.6	286.52	226.83
		6	SWB	65,305.6	240.91	262.24	65,305.6	262.24	240.91
Mandrill (Baboon)512×512	RGB	1	MT	8.2840e+04	287.62	253.10	8.2840e+04	287.62	253.10
		2	dSFMT	8.2840e+04	265.26	255.1	8.2840e+04	265.26	255.1
		3	MRG	8.2840e+04	250.95	232.6	8.2840e+04	250.95	232.6
		4	MLFG	8.2840e+04	241.12	247.7	8.2840e+04	247.7	241.12
		5	shr3cong	8.2840e+04	316.13	242.4	8.2840e+04	242.4	316.13
		6	SWB	8.2840e+04	258.89	274.2	8.2840e+04	274.2	258.89
Peppers512x512	RGB	1	MT	213,187.2	269.66	222.8	213,187.2	269.66	222.8
		2	dSFMT	213,187.2	264.13	276.5	213,187.2	276.5	264.13
		3	MRG	213,187.2	243.76	285.7	213,187.2	285.7	243.76
		4	MLFG	213,187.2	288.24	263.1	213,187.2	288.24	263.1
		5	shr3cong	213,187.2	289.09	286.95	213,187.2	289.09	286.95
		6	SWB	213,187.2	239.94	233.1	213,187.2	239.94	233.1

the comparison of the chi-square values for experiment 2, which is used for substituting each pixel in Lena Baboon, Peppers images are lower than experiment 1. Experiment 2 works to change the distribution of the pixels in the Lena, Baboon, Peppers images therefor chi-square has lower values than experiment 1. Experiment 3 gives the best chi-square values like experiment 2 results. Experiment 3 is the results of encrypting images by combining the two methods permutation and substitution. The comparisons of the best lower values in experiment 2 and experiment 3 are presented with the best chi-square values in the two experiments approximately equivalently.

4.1.3 Measures of central tendency and dispersion

Mean, median, variance, and standard deviation have been performed for comparison of performance for all the pixels in the plain and encrypted image. The comparative analysis confirms that the values for encrypted images are uniform, not alike the values for the plain image [72].

Median has computed the median value of the image. The median value of the sorted matrix is computed as Eq. 2, where N is the size of the matrix. For the odd size matrix, the middle value is the median, and for the even size matrix, the median is the mean of the middle two values.

$$\frac{N}{2} \tag{2}$$

Mean is used to calculate the brightness for the plain and encrypted images. A high mean value denotes that the image is bright, and a low mean value indicates that the image is dark. Mean is defined as Eq. 3 [55]. Where N is the total number of the pixels and x_{ji} is the pixels values.

$$\mu_j = \frac{1}{MN}\, C \sum_{j=1}^{M} \sum_{i=1}^{N} x_{ji} \tag{3}$$

Variance indicates how pixels are spread. A large variance value indicates there is changing between adjacent pixels, and it lends to the image a noise look. It is defined as Eq. 4, Where μ_j is the mean value.

$$V = \frac{1}{MN} \sum_{j=0}^{N-1} \sum_{i=0}^{M-1} \left(x_{ji} - \mu_j\right)^2 \tag{4}$$

Standard deviation is used to compute the contrast of the pixel's intensity. A high standard deviation value means that the image is high contrast, and a low standard deviation value indicates that the image is low contrast. It is defined as Eq. 5 [55].

$$\sigma_j = \sqrt{\frac{1}{MN} \sum_{i=0}^{255} \left(x_{ji} - \mu_j\right)^2} \tag{5}$$

Table 4 is studied the statistical structure of the original image, and for the original image after applying encryption with different three methods. Mean, median, variance, and standard deviation are collaborating to produce a clarification for changing the behavior of the image after encrypted.

Table 4 Measures of central tendency and dispersion

Image	Measures	Plain image	Cipher process						
			Experimental 1						Experimental 2
			Transposition						Substitution
			MT	dSFMT	MRG	MLFG	shr3cong	SWB	MT
Lena 256	Mean	128	127.9	127.9	127.9	127.9	127.9	127.9	128
	Median	131	131	131	131	131	131	131	128
	Variance	2071	2070.5	2070.5	2070.5	2070.5	2070.5	2070.5	5463
	standard deviation	44.7	44.7	44.75	44.75	44.75	44.75	44.75	73.9
Baboon	Mean	126.4	126.5	128.9	126.5	126.5	126.5	126.5	127.32
	Median	120.3	120.3	120.3	120.3	120.3	120.3	120.3	127
	Variance	3056.3	3056.3	3056.3	3056.3	3056.3	3056.3	3056.3	5457.9
	standard deviation	54.99	54.99	54.99	54.99	54.99	54.99	54.99	73.9
Peppers	Mean	110.64	110.6	110.64	110.64	110.64	110.64	110.64	127.59
	Median	114.7	114.7	114.7	114.7	114.7	114.7	114.7	128
	Variance	3211.3	3211.3	3211.3	3211.3	3211.3	3211.3	3211.3	5457.7
	standard deviation	54.8	54.8	54.8	54.8	54.8	54.8	54.8	73.9

Image	Cipher process										
	Experimental 2					Experimental 3					
	Substitution					Combination of two methods					
	dSFMT	MRG	MLFG	shr3cong	SWB	MT	dSFMT	MRG	MLFG	shr3cong	SWB
Lena 256	127.0	127.7	127.4	127.2	127.4	127.5	127.5	127.5	127.60	127.6	127.5
	126.7	128	127.7	126.7	127	127.3	127.7	127.7	127.7	128	127.3
	5458.9	5463.8	5463.5	5474.6	5474.8	5462.8	5461.7	5478.7	5452.7	5464.3	5457.6
	73.9	73.92	73.92	73.99	73.99	73.91	73.9	74.02	73.8	73.9	73.9
Baboon	127.7	127.5	127.3	127.4	127.5	127.5	127.6	127.6	127.5	127.7	127.6

Table 4 (continued)

Image	Cipher process										
	Experimental 2					Experimental 3					
	Substitution					Combination of two methods					
	dSFMT	MRG	MLFG	shr3cong	SWB	MT	dSFMT	MRG	MLFG	shr3cong	SWB
Peppers	127.7	127.3	127	127	127.7	127.3	128	127.3	128	128	128
	5449.9	5463.8	5459.9	5469.8	5464.2	5452.4	5450.8	5465.3	5455.0	5472.8	5471.5
	73.8	73.9	73.9	73.96	73.92	73.84	73.8	73.9	73.9	73.98	73.97
	127.5	127.3	127.6	127.6	127.4	127.5	127.6	127.7	127.4	127.4	127.6
	127.3	127	128	127.7	127.7	128	127.7	128	127	127.3	127.7
	5460.6	5447.9	5466.6	5464.6	5459.3	5465.6	5455.3	5461.1	5462.9	5468.3	5458.6
	73.9	73.8	73.9	73.9	73.9	73.9	73.9	73.9	73.9	73.9	73.9

The obtained remarkable results from Table 4 show that experiment 1 is the same original image statistically, its results confirm there is no change in pixels value. Whereas the comparative analysis for experiment 2 and experiment 3 show that the values for three cipher images are uniform. Furthermore, the mean, median, variance, and standard deviation values for cipher Lena, Baboon, and Peppers are similar and unlike the mean, median, variance, and standard deviation values for the original image.

For comparison of performance, every single PRNG used in experiment 1 shows that all the measure of central tendency has identical values like the values for the plain image. Experiment 2 and experiment 3 results are the opposite experiment 1 results. The results for each generator among the six PRNGs are uniform according to each term of the measure of central tendency, and its results different from than results of the original image.

4.1.4 Entropy (S)

Entropy is one of the important 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 the intensity of each pixel in color levels. Entropy can measure the information in images whether in the 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 appear. The change will be noticeable when you change pixel values or when you change pixel positions. Using encryption such as XOR or scrambling encryption helps to hide information and maintains randomness. Its reflection of 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 good image encryption. The entropy is the summation of all the possible occurrences of probability distribution pixels p_i, as shown in the following Eq. 2 [81].

$$\mathrm{H}\,(I) = -\sum_{i=1}^{256} p_i \log p_i \tag{6}$$

Table 5 shows the measuring process of data predictability or the information predictability from the ciphered image into three samples of images Lena, Baboon, and Peppers images at three stages of encryption. The first stage is the scrambling operations, the second stage is XOR encryption, and the three-stage is combining two methods of permutation and substitution, respectively. The results we got from the entropy computation was very close to 8 in experiment 2, and experiment 3, it is confirmed the good performance. This refers to the strength of the two encryption methods to resist entropy attacks. The max function denotes the ideal value of the entropy among the three encryption methods for each of the six PRNG and produced the highest value. The min function indicates the worst value of the entropy among the three encryption methods for each of the six PRNG and produced the lower value. And the moderate value of the entropy can be measured by med function for the three encryption methods according to each of the six PRNG.

Table 5 appears the best entropy results we have obtained after the image is encrypted with two methods substitution, and the combination of two methods of permutation and substitution, respectively. The range of the best results of two experiments 2 and 3 between 7.9971 to 7.9995. By focusing on the results of the Lena image, the result of the entropy indicates that the PRNG that achieved the highest number is Multiplicative Lagged Fibonacci (MLFG) equal

Table 5 Entropy measure for three enrryption methods

Images	Algo	PRNG	Plain	Cipher process			Functions		
				Experimental 1 Transposition	Experimental 2 Substitution	Experimental 3 Combination two methods	Max	Med	Min
Lena 256×256	1	MT	RGB	7.2404	7.9972	7.9973	7.9973	7.9972	7.2404
	2	dSFMT	RGB	7.2404	7.9972	7.9969	7.9972	7.9969	7.2404
	3	MRG	RGB	7.2404	7.9972	7.9971	7.9972	7.9971	7.2404
	4	MLFG	RGB	7.2404	7.9974	7.9975	7.9975	7.9974	7.2404
	5	shr3cong	RGB	7.2404	7.9974	7.9972	7.9974	7.9972	7.2404
	6	SWB	RGB	7.2404	7.9973	7.9971	7.9973	7.9971	7.2404
Mandrill512x512 (Baboon)	1	MT	RGB	7.6444	7.9993	7.9993	7.9993	7.9993	7.6444
	2	dSFMT	RGB	7.6444	7.9993	7.9993	7.9993	7.9993	7.6444
	3	MRG	RGB	7.6444	7.9993	7.9993	7.9993	7.9993	7.6444
	4	MLFG	RGB	7.6444	7.9993	7.9993	7.9993	7.9993	7.6444
	5	shr3cong	RGB	7.6444	7.9993	7.9993	7.9993	7.9993	7.6444
	6	SWB	RGB	7.6444	7.9993	7.9992	7.9993	7.9992	7.6444
Peppers512x512	1	MT	RGB	7.2977	7.9993	7.9993	7.9993	7.9993	7.2977
	2	dSFMT	RGB	7.2978	7.9993	7.9993	7.9993	7.9993	7.2978
	3	MRG	RGB	7.2978	7.9993	7.9994	7.9994	7.9993	7.2978
	4	MLFG	RGB	7.2978	7.9992	7.9993	7.9993	7.9992	7.2978
	5	shr3cong	RGB	7.2978	7.9993	7.9993	7.9993	7.9993	7.2978
	6	SWB	RGB	7.2978	7.9994	7.9993	7.9994	7.9993	7.2978

7.9975 for combination method encryption. The results of the encrypted Baboon image in both experiments 2, and experiments 3 are equal. The results of encrypted Peppers image in both experiments 2, and experiments 3 are equally for Mersenne Twister (MT), SIMD oriented Fast Mersenne Twister (dSFMT), and Shift-register generator summed with linear congruential generator (shr3cong), whereas the assessment showed that the generator with the highest value in entropy results for Peppers image is Combined multiple recursive (MRG) in experiment 3 with a value of 7.9974 and the highest value in entropy results for Peppers image is Modified subtract with borrow generator (SWB) in experiment 2 with a value of 7.9974.

4.2 Correlation-based measures

4.2.1 Correlation coefficient (CC)

In the correlation coefficient image test, it is implemented order to examine the relationship of the original image with encrypted image. In this test, the original image pixels are measured with the encrypted images pixels. 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 Eq. 7 as shown below [51]:

$$CC = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\left(A_{ij}-\overline{A}\right)\left(B_{ij}-\overline{B}\right)}{\sqrt{\left(\sum_{i=1}^{M}\sum_{j=1}^{N}\left(A_{ij}-\overline{A}\right)^{2}\right)\left(\sum_{i=1}^{M}\sum_{j=1}^{N}\left(B_{ij}-\overline{B}\right)^{2}\right)}} \tag{7}$$

A and B represent plain and cipher images, respectively. $\overline{A}$ and $\overline{B}$ are represented the mean value of the two matrices A and B. M and N refer to the high and width of the two images plain/cipher images [51]. In quantitative measure, low-value results in correlation coefficient mean good security for encrypting image, and inversely high-value results indicate high similarity between two images.

The results are listed in Table 5, which illustrates the computing of the correlation coefficient for the three samples of Lena, Baboon, and Peppers images with encrypted images by three operations of encryption. The results show that the three processes of encryption images are variance in CC results among them. All processes of encryptions are achieved approximately CC value very close to 0 in the three-way, differently. This means that the similarity between the original image and the encrypted image is nonexistent. The three functions of the max, med, and min respectively show the worst value with faraway from 0 value, the moderate value, and the ideal value, which is closer to 0 value, respectively.

To clarify the correlation coefficient results from Table 6, the results of the processes we obtained to test the Lina image after implementing three operations of encryption separately were high in expermint1 by using three generators separately. The generators scored the highest ideal value for Lena image among the three encryption process are MT, dSFMT, and MRG. In experiment 2 the highest CC value scored for substituting by using MLFG generator. And In experiment 3 the highest CC value scored for combining two methods by using shr3cong and SWB generators.

The comparison for Baboon image among the three encryption processes produced the generators with the highest values close to zero. In experiment 1 the efficient generators with high CC results are are dSFMT, and MRG with values of 0.0001, and − 0.0004. Also, dSFMT, MRG, and

Table 6 Correlation coefficient for three encryption methods

Images	Algo	PRNG	Plain	Cipher process			Functions		
				Experimental 1 Transposition	Experimental 2 Substitution	Experimental 3 Combination two methods	Max	Med	Min
Lena 256×256	1	MT	RGB	−0.0002	0.0014	−0.0027	−0.0027	0.0014	−0.0002
	2	dSFMT	RGB	−0.0029	−0.0056	−0.0068	−0.0068	−0.0056	−0.0029
	3	MRG	RGB	−0.0006	−0.0040	−0.0019	−0.0040	−0.0019	−0.0006
	4	MLFG	RGB	0.0020	0.0003	0.0033	0.0033	0.0020	0.0003
	5	shr3cong	RGB	0.0033	0.0021	−0.00007	0.0033	0.0021	0.0021
	6	SWB	RGB	0.0053	0.0028	0.0007	0.0053	0.0028	0.0007
Mandrill (Baboon)512×512	1	MT	RGB	−0.0014	0.0009	0.0005	−0.0014	0.0009	0.0005
	2	dSFMT	RGB	−0.0006	0.0001	0.0002	−0.0006	0.0002	0.0001
	3	MRG	RGB	0.0021	0.00004	0.0003	0.0021	0.0003	0.00004
	4	MLFG	RGB	0.0001	0.0027	0.0016	0.0027	0.0016	0.0001
	5	shr3cong	RGB	−0.0004	0.0025	−0.0004	0.0025	–	−0.0004
	6	SWB	RGB	0.0005	0.0001	−0.0008	−0.0008	0.0005	0.0001
Peppers512x512	1	MT	RGB	0.0002	−0.0001	−0.0011	−0.0011	−0.0011	−0.0011
	2	dSFMT	RGB	−0.0013	0.00003	−0.0007	−0.0013	−0.0007	0.00003
	3	MRG	RGB	0.0008	−0.0029	0.0002	−0.0029	0.0008	0.0002
	4	MLFG	RGB	0.0015	0.0017	−0.0001	0.0017	0.0015	−0.0001
	5	shr3cong	RGB	9.66e-04	0.000095	0.0015	0.0015	9.66e-04	0.000095
	6	SWB	RGB	0.0023	0.0001	−0.0009	0.0023	−0.0009	0.0001

Table 7 Pearson's correlation coefficient for encrypting three images based on six PRNGs

Images	Algo	PRNG	Plain	Cipher process		
				Experimental 1 Transposition	Experimental 2 Substitution	Experimental 3 Combination two methods
Lena 256×256	1	MT	RGB	−15.1	94.16	−178.14
	2	dSFMT	RGB	−191.7	−365.6146	−444.22
	3	MRG	RGB	−42.04	−261.830	−121.6
	4	MLFG	RGB	132.7	21.79	217.37
	5	shr3cong	RGB	213.7	136.04	−5.09
	6	SWB	RGB	348.03	184.08	47.4
Mandrill (Baboon)512× 512	1	MT	RGB	−365.67	233.68	153.01
	2	dSFMT	RGB	−164.8	39.02	47.67
	3	MRG	RGB	551.8	11.34	67.52
	4	MLFG	RGB	30.34	703.69	407.1
	5	shr3cong	RGB	−117.7	666.74	−122.1
	6	SWB	RGB	1.43	34.31	−224.1
Peppers512x512	1	MT	RGB	78.51	−29.59	−279.5
	2	dSFMT	RGB	−340.7	8.22	−203.96
	3	MRG	RGB	221.2	−758.96	63.95
	4	MLFG	RGB	401.3	435.94	−51.01
	5	shr3cong	RGB	253.2	24.93	405.7
	6	SWB	RGB	591.2	39.03	−240.5

SWB with values of 0.0001, 0.00004, and 0.0001 in experiment 2 respectively. Whereas, in experiment 3 the efficient generators with high CC results are MT with a value of 0.0005.

The results we obtained from the test to investigate Peppers image after implementing three operations of encryption separately, were between interval 0.00003 and − 0.0029. The generator with the highest value close to zero among three prosses for dSFMT, shr3cong, and SWB are in experiment 2 with values equal 0.00003, 0.000095,and 0.0001, respectively.

A. Pearson's Correlation Coefficient (PCC)

The Pearson's correlation coefficient is commonly utilized in pattern detection, mathematic analysis, and image processing. For the later, application include a comparison of two images for the reason object recognition, image recoding, and measurement of disparity [85]. Mathematically, the Pearson correlation coefficient is computed as Eq. 8.

$$PCC_{xy} = \frac{cov(x,y)}{\sigma_x \sigma_y}$$
$$\text{Where } cov(x,y) = \sum_{i=1}^{M}\sum_{j=1}^{N}\left(X_{ij}-\overline{X}\right)\left(Y_{ij}-\overline{Y}\right) \tag{8}$$
$$\sigma_x = \sum_{i=1}^{M}\sum_{j=1}^{N}\left(X_{ij}-\overline{X}\right) \text{ and } \sigma_y = \sum_{i=1}^{M}\sum_{j=1}^{N}\left(Y_{ij}-\overline{Y}\right)$$

Where X_{ij} indicates the intensity of the coordinate pixel in the original image, Y_{ij} indicates the intensity of the coordinate pixel in an encrypted image, $\overline{X}$ and $\overline{Y}$ is the mean intensity of original, and encrypted images, respectively. A lower value means the similarity between the two images does not exist, whereas a higher value indicates the two images are identical. For a comparison of performance, the following Table 7 shows the differences between the six PRNGs among three processes of encryption.

Table 8 Correlation of adjacent pixels for encrypting three images by using six of PRNGs

Image	No.	PRNG	Cipher process								
			Experimental 1			Experimental 2			Experimental 3		
			Transposition			Substitution			Combination two methods		
			Orientation			Orientation			Orientation		
			Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
Lena 256×256	1	MT	0.0124	0.0110	0.0116	−0.013	0.015	−0.006	0.0196	0.0003	−0.0023
	2	dSFMT	0.0147	−0.0072	0.0057	−0.0089	−0.0124	0.0050	−0.0262	0.0015	−0.0006
	3	MRG	0.0177	−0.0182	0.0200	0.0225	−0.0081	0.0171	0.0083	0.0031	−0.0067
	4	MLFG	−0.0261	−0.0051	−0.0026	−0.0081	0.0116	0.0197	0.0061	0.0283	−0.0155
	5	shr3cong	−0.0232	−0.0232	−0.0175	−0.0129	−0.0067	0.0284	−0.0102	0.0041	0.0306
	6	SWB	−0.0038	0.0176	0.0029	−0.0038	−0.0176	−0.00044	−0.0023	0.0454	−0.0103
Mandrill (Baboon)512×512	1	MT	−0.0061	0.0135	0.0235	−7.0e-04	−0.0012	−0.0128	0.0136	−0.0146	−0.0087
	2	dSFMT	0.0255	−0.0270	0.0150	0.0153	−0.0162	0.0147	0.0020	−0.0003	−0.0079
	3	MRG	−0.0240	0.0026	−0.0249	0.0521	0.0033	−0.0124	0.0042	−0.0049	0.0071
	4	MLFG	0.0015	−0.0029	0.0097	−0.0348	0.0008	0.0252	−0.0018	−0.0100	0.0024
	5	shr3cong	0.0290	−0.0152	−0.0272	−0.0248	0.0221	0.0192	−0.0490	−0.0138	0.0204
	6	SWB	0.0067	−0.0392	−0.0092	0.0015	−0.0006	−0.0104	0.0049	−0.0080	−0.0048
Peppers512x512	1	MT	0.0412	0.0362	0.0101	0.0137	−0.0018	0.0378	−0.0178	0.0037	0.0200
	2	dSFMT	0.0306	0.0287	0.0123	0.0014	−0.0677	0.0143	0.0110	−0.0179	−0.0014
	3	MRG	−0.0051	0.0022	0.0034	0.0285	0.0065	−0.0024	0.0073	−0.0389	0.0177
	4	MLFG	−0.0352	0.0278	0.0027	0.0032	−0.0174	0.0007	−0.0014	0.0090	0.0057
	5	shr3cong	0.0328	−0.0166	−0.0054	−0.0464	−0.0051	0.0354	0.0215	0.0497	−0.0162
	6	SWB	−0.0272	0.0317	−0.0096	0.0019	−0.0239	−0.0110	−0.0192	−0.0263	−0.0067

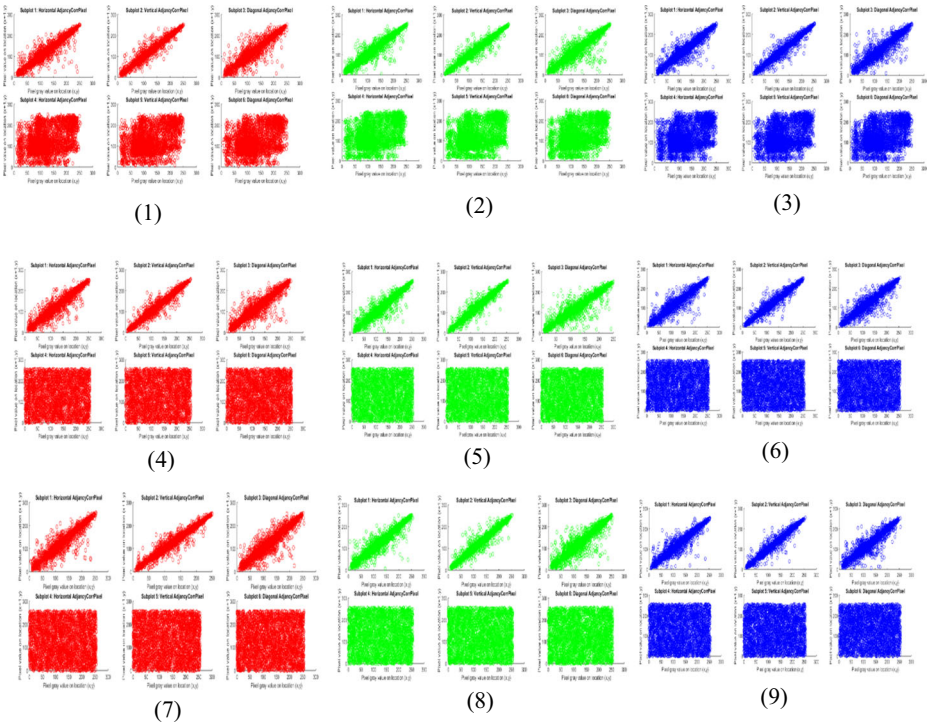


Fig. 20 Correlation of adjacent pixels: (1–3) horizontal, vertical, and diagonal correlation for each channel of plain Lena and scrambled image using MT generator. (4–6) horizontal, vertical, and diagonal correlation for each channel of plain Lena and XORed image using MT generator. (7–9) horizontal, vertical, and diagonal correlation for each channel of plain Lena and encrypted image by combining of two encryption processes using MT generator

B. Correlation of Adjacent Pixels

Analysis of the correlation of adjacent pixels in encrypted images is very important to test their relationship between adjacent pixels. We selected randomly 3000 pairs of adjacent pixels from original and encrypted images. A high correlation of Adjacent Pixels appears in the original image and has a value close to ±1. Whereas a low correlation should score for the ciphered image with a value close to 0. The good encryption method should destroy the strong correlation among adjacent pixels in the original image to resist statistical attacks. The following equations explain the calculation of the correlation coefficient for each pair of two adjacent pixels r_{xy}, where x and y represent the gray-scale value of two adjacent pixels in the image, and N denotes the entire number of pixels picked from the image [43]. Table 8 presents the result for encrypting images with three processes by using six PRNGs, separately. Figure 20 displays the strong correlation for the original image in three orientations, and the correlation of encrypted image in three processes of encryption with three orientations.

$$E(x) = \frac{1}{N} \sum_{i=1}^{N} x_i \tag{9}$$

$$D(x) = \frac{1}{N} \sum_{i=1}^{N} (x_i - E(x))^2 \tag{10}$$

$$cov(x,y) = \frac{1}{N} \sum_{i=1}^{N} (x_i - E(x))(y_i - E(y)) \tag{11}$$

$$r_{xy} = \frac{cov(x,y)}{\sqrt{D(x).D(y)}} \tag{12}$$

C. Normalized cross-correlation calculation (NCC)

A normalized cross-correlation (NCC) is a metric to evaluate the difference between two digital images, it might be original and encrypted digital images. We used NCC to measure the degree of correlation between the two images. The range of results from normalized cross-correlation is between −1 and 1. If the NCC is considered as 1, this means the difference between the two images does not exist. For each image, the calculation of similarity depends on the direct splendor and complication of the cross-correlation varieties. Normalized cross-correlation is identified as Eq. 13, where m × n indicates the size of both original image P and encrypted image C [86].

$$NCC = \frac{\sum_{j=1}^{m} \sum_{k=1}^{n} P_{jk} \times C_{jk}}{\sum_{k=1}^{n} \left(P_{jk}\right)^2} \tag{13}$$

Table 9 presents the results of Lena, Baboon, and Peppers images according to three cipher processes by using different types of PRNGs, separately. The best and ideal value of NCC is presented in the min function for each PRNG among the three processes of encryption. The med function denotes moderate results. The max function indicates the worst results of distinction between the original and ciphered images.

4.3 Image quality assessment (IQA) metrics

4.3.1 Image error and quality measurements

We can determine the image quality by studying the following measures: mean square error (MSE), root mean square error (RMSE), mean absolute error (MAE), peak signal to noise ratio (PSNR), Signal to Noise Ratio (SNR), structural similarity index (SSIM), and mutual information (MI).

1. Mean square error (MSE): MSE computes the differences squared between the whole pixel image, then divides it by the total number of pixels. It can be calculated as equation 54 [9]. If the MSE value appears to be zero, this means that the highest value of the similarity of the two images has been achieved. The opposite represents the dissimilarity of the two images.

Table 9 Normalized cross correlation calculation for encrypting three images by using six of PRNGs

Original Images With *NCC*	PRNG No.	PRNG Algo	Plains	Encrypted image by three cipher process			Functions		
				Experimental 1 Transposition NCC	Experimental 2 Substitution NCC	Experimental 3 Combination of two methods NCC	Max	Med	Min
Lena 256×256 (6.5306e+04)	1	MT	RGB	0.8737	0.9245	0.9242	0.9245	0.9242	0.8737
	2	dSFMT	RGB	0.8734	0.9202	0.9234	0.9234	0.9202	0.8734
	3	MRG	RGB	0.8737	0.93	0.9246	0.93	0.9246	0.8737
	4	MLFG	RGB	0.8741	0.9236	0.9261	0.9261	0.9236	0.8741
	5	shr3cong	RGB	0.8742	0.9222	0.9253	0.9253	0.9222	0.8742
	6	SWB	RGB	0.8745	0.9235	0.9243	0.9243	0.9235	0.8745
Mandrill (Baboon)512×512	1	MT	RGB	0.8362	0.8444	0.8456	0.8456	0.8444	0.8362
	2	dSFMT	RGB	0.8364	0.8466	0.8464	0.8466	0.8464	0.8364
	3	MRG	RGB	0.8368	0.8453	0.8459	0.8459	0.8453	0.8368
	4	MLFG	RGB	0.8366	0.8450	0.8460	0.8460	0.8450	0.8366
	5	shr3cong	RGB	0.8366	0.8453	0.8465	0.8465	0.8453	0.8366
	6	SWB	RGB	0.8366	0.8457	0.8457	0.8457	0.8457	0.8366
Peppers512x512	1	MT	RGB	0.7711	0.9622	0.9611	0.9622	0.9611	0.7711
	2	dSFMT	RGB	0.7707	0.9618	0.9615	0.9618	0.9615	0.7707
	3	MRG	RGB	0.7712	0.9594	0.9630	0.9630	0.9594	0.7712
	4	MLFG	RGB	0.7713	0.9620	0.9606	0.9620	0.9606	0.7713
	5	shr3cong	RGB	0.7713	0.9618	0.9611	0.9618	0.9611	0.7713
	6	SWB	RGB	0.7715	0.9609	0.9618	0.9618	0.9609	0.7715

Table 10 Image error and quality measurements for encrypting three image by using six of PRNGs

Image	Measures	Cipher process															
		Experimental 1							Experimental 2						Experimental 3			
		Transposition							Substitution						Combi nation two methods of encryption			
		MT	dSFMT	MRG	MLFG	shr3cong	SWB	MT	dSFMT	MRG	MLFG	shr3cong	SWB	MT	dSFMT	MRG	MLFG	shr3c
Lena	MSE	4141.5	4152.9	4142.3	4131.9	4126.9	4119.1	8868.718	8949.8	8.93e+03	8907.85	8881.05	8878.81	8927.13	8966.2	8948.4	8881.3	8888
	RMSE	64.35	64.44	64.36	64.28	64.24	64.18	94.174	94.6036	94.52	94.38	94.24	94.23	94.48	94.69	94.60	94.24	94.28
	MAE	50.44	50.45	50.42	50.30	50.35	50.26	77.250	77.6525	77.614	77.429	77.360	77.374	77.594	77.776	77.706	77.325	77.34
	PSNR	12.27	12.26	12.2660	12.2776	12.2830	12.2919	8.7084	8.6755	8.6785	8.6926	8.7047	8.7071	8.6820	8.6683	8.6730	8.7053	8.699
	SNR	6.5392	6.527	6.5374	6.5490	6.5544	6.5633	3.9494	3.8881	3.9289	3.9237	3.9286	3.9400	3.9229	3.9050	3.9153	3.9462	3.946
	NAE	0.4211	0.4211	0.4208	0.4196	0.4201	0.4194	0.6410	0.6435	0.6437	0.6422	0.6416	0.6416	0.6436	0.6445	0.6442	0.6412	0.641
Baboon	MSE	6122.1	6117.8	6100.1	6110.7	6113.3	6109.0	8608.208	8607.8	8624.2	8597.6	8608.0	8622.3	8607.3	8610.8	8.6210	8601.3	8635
	RMSE	78.24	78.22	78.10	78.17	78.19	78.16	92.780	92.778	92.867	92.723	92.780	92.856	92.77	92.79	92.85	92.74	92.92
	MAE	62.78	62.75	62.68	62.73	62.76	62.73	76.257	76.271	76.348	76.214	76.306	76.375	76.28	76.28	76.35	76.25	76.44
	PSNR	10.3563	10.3596	10.37	10.36	10.3604	10.3647	8.7964	8.7967	8.7885	8.8019	8.7968	8.7892	8.7965	8.7950	8.7897	8.7999	8.782
	SNR	5.0184	5.0217	5.0336	5.0250	5.0225	5.0269	4.0240	4.0395	4.0247	4.0306	4.0307	4.0285	4.0311	4.0368	4.0310	4.0374	4.031
	NAE	0.5021	0.5019	0.5014	0.5018	0.5019	0.5017	0.6082	0.6083	0.6089	0.6078	0.6086	0.6091	0.6082	0.6084	0.6089	0.6081	0.609
Peppers	MSE	6423.0	6431.9	6413.8	6415.4	6414.5	6407.6	10,123.5	10,128.8	10,130.6	10,111.4	10,128.8	10,114.5	10,136.2	10,138.1	10,130.3	10,126.2	10,11
	RMSE	80.14	80.02	80.09	80.10	80.09	80.05	100.62	100.642	100.65	100.56	100.64	100.57	100.7	100.7	100.6	100.6	100.6
	MAE	61.40	61.42	61.31	61.32	61.31	61.30	82.183	82.225	82.262	82.126	82.216	82.225	82.26	82.28	82.25	82.25	82.14
	PSNR	10.6119	10.6050	10.6143	10.6173	10.6148	10.6204	8.1297	8.1278	8.1263	8.1349	8.1272	8.1328	8.1245	8.1244	8.1268	8.1284	8.135
	SNR	4.0571	4.0501	4.0594	4.0624	4.0599	4.0656	3.3709	3.3672	3.3546	3.3759	3.3682	3.3655	3.3635	3.3640	3.3720	3.3624	3.371
	NAE	0.5994	0.5997	0.5986	0.5986	0.5986	0.5986	0.8456	0.8461	0.8462	0.8449	0.8458	0.8460	0.8461	0.8468	0.8461	0.8461	0.845

$$MSE = \frac{1}{MN} \sum_{X=0}^{M-1} \sum_{Y=0}^{N-1} \left(C(x,y) - P\left(x,y\right) \right)^2 \tag{14}$$

2. Root mean square error (RMSE): The small value from RMSE informs that the original and encrypted images are more similar. Whereas the bigger RMSE indicates that the similarity between the two images is different, it can be calculated as Eq. 15, where MN refers to the size of the image. Also, R(i,j) and I(i,j) denote the original image and the encrypted images.

$$RMSE = \sqrt{\frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} |R(i,j) - I(i,j)|^2} \tag{15}$$

3. Mean absolute error (MAE): MAE is defined as Eq. 16. C(x,y) and P(x,y) indicate cipher and plain images. MN indicates to size image. The higher MAE is better in good encryption [51].

$$MAE = \frac{1}{MN} \sum_{y=1}^{M} \sum_{x=1}^{N} \left| \left(C(x,y) - P\left(x,y\right) \right| \right| \tag{16}$$

4. Peak signal to noise ratio (PSNR): 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 quality of the image is good and the lower value means the whole image is noise. It can be described as Eq. 17.

$$PSNR = 10 log_{10} \left(\frac{255^2}{MSE} \right) \tag{17}$$

5. Signal to Noise Ratio (SNR): SNR is computed as the ratio of the signal power to the noise power. Low-value results obtained from SNR indicated that the encrypted image is more secure. The SNR is expressed as Eq. 18, where p_s denotes to signal power and p_n represents noise power.

$$SNR = 10 \log_{10} \frac{p_s}{p_n} \tag{18}$$

6. Normalized Absolute Error (NAE): NAE is calculated as Eq. 19, it's the sum of the absolute difference between original and encrypted images over the sum of the original image. A higher NAE calculation reveals the wonderful value of the obtained encrypted image after the encryption process.

$$NAE = \frac{\sum_{j=1}^{m}\sum_{k=1}^{n} |P_{jk} - C_{jk}|}{\sum_{j=1}^{m}\sum_{k=1}^{n} P_{jk}} \tag{19}$$

The following Table 10 shows the results of quality measures for encrypting three images by three operations of encryption. All metrics applied between the original images with the encrypted image. The result of Peak signal to noise ratio (PSNR), and Signal to Noise Ratio (SNR) are low as should be for totally dissimilar images. Mean square error (MSE), root mean square error (RMSE), Mean absolute error (MAE), and Normalized Absolute Error (NAE) are very high.

4.3.2 Human visual system (HVS) features

1. Structural similarity index (SSIM): SSIM is used to calculate the similarity between two images. The value results of SSIM statistics range from −1 and 1. The high value in SSIM means that the two sets of data are identical, and the two comparative images are the same. The opposite value means that the two images are totally different. The SSIM metric is measured as shown in Eq. 20, where $\mu_{\S}$ denotes to the average of x, and $\mu_y$denotes to the average of y. σ_x^2represents the variance of x, and σ_y^2 refers to the variance of y. $C_1$and C_2 are two variables to achieve the stabilization of the division with a weak denominator[1].

$$SSIM\ (x,y) = \frac{\left(2\mu_{\S}\mu_y + C_1\right)\left(2\sigma_{\S y} + C_2\right)}{\left(\mu_x^2 + \mu_y^2 + C_1\right)\left(\sigma_x^2 + \sigma_y^2 + C_2\right)} \tag{20}$$

2. Structural dissimilarity (DSSIM): DSSIM is a measure derived from SSIM. It is used to analyze the dissimilarities between original and encrypted images. A high DSSIM value means that the two images are dissimilar, and a low DSSIM value confirms that the two images are the same. DSSIM measure is defined as Eq. 21 [16]:

$$DSSIM(x,y) = \frac{1 - SSIM(x,y)}{2} \tag{21}$$

3. Multi-scale structural similarity index (MS-SSIM):

MS_SSIM is proposed by Wang et al. [79]. The term refers to an evaluation of the image at different scales [56]. It is applied on multiple scales of the original and encrypted images. MSSIM returns a numeric value between 0 and 1. The highest quality for the image accomplishes 1 in MS-SSIM index. MSSIM is calculated as Eq. 22, where $I_m\ (x,y)$, $C_i\ (x, y)$, and $S_i(x,y)$ are multiple scales of contrast, structure, and luminance. α, β, and y are parameters which are selected as $\alpha_i = \beta_i = \gamma_i$ and $\sum_{i=1}^{m} \gamma_i = 1$ [34].

$$MSSIM\ (x,y) = [(I_m\ (x,y))]^{\alpha_m} \prod_{i=1}^{m} [(C_i\ (x,y))]^{\beta_i}.\ [(S_i(x,y))]^{\gamma_i} \tag{22}$$

4. Universal Image Quality Index (UIQI): UIQI presented by Wang and Bovik in 2002 [78]. The comparison between original and encrypted image is broken into luminance l(x, y), contrast c(x, y), and structural comparisons s(x, y) such as three Eqs. in (23), (24) and (25).

$$l(x,y) = \frac{2\mu_x\mu_y}{\mu_x^2 + \mu_y^2} \tag{23}$$

$$c(x,y) = \frac{2\sigma_x\sigma_y}{\sigma_x^2 + \sigma_y^2} \tag{24}$$

$$s(x,y) = \frac{2\sigma_{xy}}{\sigma_x + \sigma_y} \tag{25}$$

Where $\mu_x\mu_y$, $\sigma_x\sigma_y$, and σ_{xy} denote the mean values, the standard deviation, and the covariance of original and encrypted images, respectively. Based on the above three equations the UIQI is given in Eq. 26 [6].

$$UIQI = \mathrm{l(x,y).c(x,y).s(x,y)} = \frac{4\mu_x\mu_y\mu_{xy}}{\left(\mu_x^2 + \mu_y^2\right)\sigma_x^2 + \sigma_y^2} \tag{26}$$

UIQI has a range between −1 and 1. The value 1 denotes that the original and encrypted images are similar.

The following Table 11 shows the results of quality measures for encrypting three images by three operations of encryption. All metrics applied between the original images with the encrypted image. Structural similarity index (SSIM), Multi-scale structural similarity index (MS-SSIM), and Universal Image Quality Index (UIQI) are nearly 0 for all images- which means that the similarity between images and their encrypted versions is nonexistent. Lastly, the Structural dissimilarity (DSSIM) is high, which means that there is no similarity between the original image and its encrypted image.

1. IQMs based on difference distortion and statistical pixel distance

A. Image Fidelity (IF): The fidelity calculates the proximity of an image to its typical image. Image fidelity is defined as Eq. 27 [23].

Table 11 Human Visual System (HVS) features

Image	Measures	Channel	Cipher process						
			Experimental 1						Experimental 2
			Transposition						Substitution
			MT	dSFMT	MRG	MLFG	shr3cong	SWB	MT
Lena	SSIM	RGB	0.0309	0.0282	0.0314	0.0312	0.0290	0.0303	0.0094
	DSSIM	RGB	0.4846	0.4859	0.4843	0.4844	0.4855	0.4848	0.4953
	MS_SSIM	RGB	0.0309	0.0286	0.0312	0.0310	0.0472	0.0302	0.0093
	UIQI	RGB	0.0014	−0.0015	0.0021	0.0017	9.3e-06	0.0011	−0.00055
Baboon	SSIM	RGB	0.0158	0.0176	0.0155	0.0150	0.0160	0.0160	0.0096
	DSSIM	RGB	0.4921	0.4912	0.4923	0.4925	0.4902	0.4920	0.4952
	MS_SSIM	RGB	0.0156	0.0176	0.0156	0.0150	0.0162	0.0160	0.0094
	UIQI	RGB	−0.0008	0.0015	−0.0006	−0.0015	−9.3e-05	−0.0002	1.49e-04
Peppers	SSIM	RGB	0.0205	0.0200	0.0213	0.0205	0.0204	0.0208	0.0087
	DSSIM	RGB	0.4897	0.4900	0.4894	0.4898	0.4898	0.4896	0.4956
	MS_SSIM	RGB	0.0205	0.0200	0.0212	0.0205	0.0204	0.0210	0.0087
	UIQI	RGB	0.0003	−0.0004	0.0008	0.0003	4.1e-05	9.2e-04	−0.0001

Image	Cipher process										
	Experimental 2					Experimental 3					
	Substitution					Combination of two methods					
	dSFMT	MRG	MLFG	shr3cong	SWB	MT	dSFMT	MRG	MLFG	shr3cong	SWB
Lena	0.0080	0.0088	0.0103	0.0104	0.0109	0.0097	0.0085	0.0090	0.0100	0.0091	0.0102
	0.4960	0.4956	0.4948	0.4948	0.4946	0.4952	0.4958	0.4955	0.4950	0.4955	0.4949
	0.0079	0.0085	0.0106	0.0102	0.0110	0.0097	0.0084	0.0090	0.0099	0.0089	0.0100
	−0.0026	−0.0015	8.26e-04	5.90e-04	0.0020	−9.3e-06	−0.0019	−0.0007	0.00005	−0.0013	0.0003
Baboon	0.0095	0.0096	0.0106	0.0102	0.0101	0.0098	0.0089	0.0094	0.0104	0.0095	0.0094

Table 11 (continued)

Image	Cipher process										
	Experimental 2					Experimental 3					
	Substitution					Combination of two methods					
	dSFMT	MRG	MLFG	shr3cong	SWB	MT	dSFMT	MRG	MLFG	shr3cong	SWB
	0.4952	0.4952	0.4947	0.4949	0.4950	0.4951	0.4956	0.4953	0.4948	0.4953	0.4953
	0.0095	0.0092	0.0105	0.0101	0.0101	0.0098	0.0090	0.0094	0.0106	0.0093	0.0095
	0.0001	−0.0002	0.0011	0.0009	0.0009	0.0005	−0.0006	−0.0002	0.0013	−0.0001	0.00001
Peppers	0.0086	0.0086	0.0096	0.0091	0.0091	0.0086	0.0087	0.0087	0.0087	0.0091	0.0086
	0.4957	0.4957	0.4952	0.4955	0.4955	0.4957	0.4956	0.4956	0.4957	0.4954	0.4957
	0.0084	0.0086	0.0096	0.0090	0.0090	0.0086	0.0086	0.0087	0.0086	0.0092	0.0086
	−0.0003	−0.0003	0.0009	0.0002	0.0002	−0.00009	−0.0002	−0.0002	−0.00002	0.0005	−0.0001

$$IF = 1 - \left(\frac{\sum_{j=1}^{M} \sum_{k=1}^{N} [P(j,k) - C(j,k)]^2}{\sum_{j=1}^{M} \sum_{k=1}^{N} \left[P(j,k)^2 \right]} \right) \tag{27}$$

Where P(j,k) is the original image, and C(j,k) is the encrypted image at j and k coordinates. M and N represent the number of rows and columns of pixels of the images.

B. Average difference (AD): AD is used to compute the noise between two images. The high AD value denotes how different is the encrypted image from the original image and its poor quality [58]. The formula of difference can be formed as the mean of an absolute of a subtract between each pixel in the original image and its corresponding in the encrypted image. AD equation represents as Eq. 28, Where X(i,j) and y(i,j) are the original and encrypted images, respectively [22].

$$AD = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} abs((x(i,j)) - (y(i,j)))}{M \times N} \tag{28}$$

To get the percentage the average difference would be calculated as the AD and is divided by 255 as represented in Eq. 29 [22].

$$AD\% = \frac{AD}{255} \tag{29}$$

C. Maximum difference (MD): MD is defined as the absolute difference between two images, the difference between the two images would be between original and encrypted images. The large value of the MD denotes that the image quality is destroyed [58]. MD is calculated as represented in Eq. 30, where x(i,j) indicates the original image, and y(i,j) denotes the encrypted image.

$$MD = Max(|x(i,j) - y(i,j)|) \tag{30}$$

D. Mean bias (MB): MB is calculated as the difference between original and encrypted images. The zero value indicates that the original and encrypted images are similar. MB is defined as Eq. 31 where x is the original image and y is the encrypted image [36].

$$MB = \frac{x_{mean} - y_{mean}}{x_{mean}} \tag{31}$$

2. The IQMs based on the association of the content of the images

   A. The structure content (SC): SC is defined as Eq. 32 between original and encrypted images. Where f(n, m) denotes an original image, and g(n, m) is the encrypted image [16]. A high value of SC means that the image quality is destroyed.

$$SC = \frac{\sum_{n=1}^{n}\sum_{m=1}^{m}[f(n,m)]^2}{\sum_{n=1}^{n}\sum_{m=1}^{m}[g(n,m)]^2} \tag{32}$$

B. Mutual information (MI): MI measurement is about how closely the two images relate. The value of MI indicates the extent to which the encrypted image contains information about the original image. A low MI value means good encryption. The following Eq. 33 shows the mathematical calculation of MI measurement, where p(x,y) denotes the probability distribution function of original and encrypted images, and p(x), p(y) indicate the probability intensity functions in the single images [16].

$$MI(X,Y) = \sum_{y\in Y}\sum_{x\epsilon X} p(x,y)\log\left(\frac{p(x,y)}{p(x)p(y)}\right) \tag{33}$$

C. Correlation Quality (CQ): Correlation quality calculates the encrypted image degradation compared to an original image depending on the size of each image. Correlation quality is defined as Eq. 34, where P_{jk} is the original image, and C_{jk} is the encrypted image [23].

$$CQ = \frac{\sum_{j=1}^{m}\sum_{k=1}^{n} P_{jk} \times C_{jk}}{\sum_{j=1}^{m}\sum_{k=1}^{n} P_{jk}} \tag{34}$$

The following Table 12 shows the results of quality measures for encrypting three images by three operations of encryption. All metrics applied between the original images with the encrypted image. The huge estimation of Average difference (AD), (AD%), Maximum difference (MD), Mean bias (MB), and the structure content (SC) imply that the three encrypted images are of low quality. Whereas the low estimation of Image Fidelity (IF), Mutual information (MI), and Correlation Quality (CQ) are indicating that the three encrypted images of low quality.

4.4 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 were able to use two of the metrics that make it happen,

Table 12 IQAs based on distinction distortion and on association of the content of the images

Image	Measures	Cipher process							
		Experimental 1						Experimental 2	
		Transposition						Substitution	
		MT	dSFMT	MRG	MLFG	shr3cong	SWB	MT	dSFMT
Lena	IF	0.7474	0.7468	0.7474	0.7482	0.7484	0.7491	0.443	0.4610
	AD	16.8162	16.82	16.81	16.77	16.78	16.75	25.750	25.8842
	AD%	0.0659	0.0660	0.0659	0.0658	0.0658	0.0657	0.101	0.1015
	MD	180.3	187.3	184	180.3	182.7	178	228.66	224.7
	MB	0.0314	0.0316	0.0316	0.0313	0.0309	0.0315	−0.0346	−0.0315
	SC	1	1	1	1	1	1	0.909	0.9175
	MI	0.2452	0.2441	0.2490	0.2444	0.2408	0.2448	0.443	0.4468
	CQ	127.9	127.8	127.8	127.9	127.9	128	127.59	126.8764
Baboon	IF	0.6725	0.6728	0.6737	0.6732	0.6731	0.6733	0.543	0.543
	AD	20.9257	20.92	20.8921	20.91	20.92	20.91	25.419	25.424
	AD%	0.0821	0.0820	0.0819	0.0820	0.0820	0.0820	0.0996	0.0997
	MD	237	236	234	233.3	234.3	238.3	243	243
	MB	0.0428	0.0426	0.0428	0.0429	0.0427	0.0429	0.0311	0.0285
	SC	1	1	1	1	1	1	0.8836	0.8807
	MI	0.1719	0.1712	0.1707	0.1726	0.1708	0.1725	0.182	0.1813
	CQ	126.41	126.4	126.5	126.5	126.5	126.5	127.35	127.7
Peppers	IF	0.5423	0.5413	0.5424	0.5427	0.5426	0.5430	0.112	0.112
	AD	20.47	20.47	20.44	20.44	20.44	20.43	27.394	27.408
	AD%	0.0803	0.0803	0.0801	0.0802	0.0801	0.0801	0.1074	0.1075
	MD	223.3	223	222.7	221.3	225.3	222	225.7	226.7
	MB	0.0277	0.0278	0.0271	0.0279	0.0278	0.0276	−0.2515	−0.2513
	SC	1	1	1	1	1	1	0.7646	0.7651
	MI	0.0985	0.0994	0.1009	0.1010	0.0993	0.0998	0.1405	0.1421
	CQ	110.6	110.6	110.7	110.7	110.7	110.7	127.60	127.54

Image	Cipher process									
	Experimental 2				Experimental 3					
	Substitution				Combination of two methods					
	MRG	MLFG	shr3cong	SWB	MT	dSFMT	MRG	MLFG	shr3cong	SWB
Lena	0.4602	0.4625	0.4638	0.4642	0.4605	0.4598	0.4604	0.4642	0.4629	0.4644
	25.87	25.81	25.787	25.79 14	25.865	25.925	25.902	25.78	25.78	25.7641
	0.1015	0.1012	0.1011	0.1011	0.1014	0.1017	0.1016	0.1011	0.1011	0.1010
	226.7	227	229.3	255	224.7	224.7	228	225.7	224.3	226.3
	−0.0365	−0.0339	−0.0321	−0.0330	−0.0345	−0.0350	−0.0350	−0.0361	−0.0357	−0.0335
	0.9089	0.9132	0.9130	0.9106	0.9112	0.9124	0.9112	0.9114	0.9083	0.9099
	0.4472	0.4399	0.4457	0.4424	0.4474	0.4473	0.4498	0.4466	0.4464	0.4478
	127.631	127.378	144.997	144.997	127.471	127.3	127.5	127.7	127.7	127.5
Baboon	0.5417	0.5431	0.5425	0.5418	0.5428	0.5424	0.5418	0.5428	0.5411	0.5408
	25.449	25.405	25.435	25.458	25.43	25.43	25.45	25.42	25.48	25.48
	0.0998	0.0996	0.0997	0.0998	0.0997	0.0997	0.0998	0.0997	0.09992	0.0999
	243.3	242	243	243.3	244.7	242.7	242.7	241.3	245.7	242
	0.0299	0.0309	0.0305	0.0296	0.0300	0.0288	0.0294	0.0295	0.0285	0.0294
	0.8820	0.8835	0.8824	0.8814	0.8823	0.8810	0.8809	0.8817	0.8794	0.8806
	0.1821	0.1820	0.1825	0.1832	0.1807	0.1811	0.1805	0.1820	0.1818	0.1802
	127.47	127.42	127.48	127.53	127.5	127.6	127.6	127.6	127.7	127.5
Peppers	0.1124	0.1137	0.1123	0.1133	0.1123	0.1107	0.1123	0.1125	0.1135	0.1132
	27.421	27.375	27.405	27.408	27.42	27.43	27.42	27.42	27.38	27.41
	0.1075	0.1074	0.1075	0.1075	0.1075	0.1076	0.1075	0.1075	0.1074	0.1075
	225.7	226.3	226.3	229.3	226.3	225.7	226	227	227.3	226.7
	−0.2492	−0.2507	−0.2511	−0.2500	−0.2502	−0.2519	−0.2520	−0.2499	−0.2499	−0.2513
	0.7669	0.7643	0.7646	0.7661	0.7648	0.7654	0.7640	0.7660	0.7652	0.7637
	0.1419	0.1408	0.1414	0.1416	0.1407	0.1409	0.1392	0.1411	0.1413	0.1416
	127.25	127.62	127.6	127.4	127.5	127.5	127.7	127.4	127.5	127.6

Table 13 NPCR and UACI measures for encrypting three images by using six types of PRNGs

Original Images	PRNG No.	PRNG Algo	Encrypted image by three cipher process					
			Experimental 1		Experimental2		Experimental 3	
			Transposition		Substitution		Combination of two methods	
			UACI%	NPCR%	UACI%	NPCR%	UACI%	NPCR%
Lena 256×256 (6.5306e+04)	1	MT	19.78	99.23	30.29	99.62	30.43	99.62
	2	dSFMT	19.78	99.23	30.45	99.58	30.50	99.61
	3	MRG	19.77	99.25	30.44	99.57	30.47	99.59
	4	MLFG	19.73	99.22	30.65	99.62	30.32	99.60
	5	shr3cong	19.74	99.28	30.34	99.65	30.33	99.62
	6	SWB	19.71	99.25	30.34	99.60	30.31	99.64
Mandrill (Baboon)512×512	1	MT	24.62	99.46	29.90	99.60	29.91	99.61
	2	dSFMT	24.61	99.45	29.91	99.91	29.91	99.63
	3	MRG	24.58	99.46	29.94	99.61	29.94	99.62
	4	MLFG	24.60	99.47	29.89	99.61	29.90	99.60
	5	shr3cong	24.61	99.46	29.92	99.63	29.98	99.60
	6	SWB	24.60	99.45	29.95	99.60	29.97	99.61
Peppers512x512	1	MT	26.58	99.10	32.23	99.60	32.60	99.60
	2	dSFMT	26.60	99.09	32.25	99.61	32.26	99.60
	3	MRG	26.54	99.11	32.25	99.61	32.25	99.61
	4	MLFG	26.55	99.10	32.21	99.62	32.25	99.61
	5	shr3cong	26.54	99.10	32.24	99.63	32.21	99.60
	6	SWB	26.54	99.10	32.24	99.60	32.25	99.62
Theoretical UACI	33.3445/33.5826	Decision	Fail	–	Fail	–	Fail	–
	33.3730/33.5541	Decision	Fail	–	Fail	–	Fail	–
Theoretical NPCR	99.5810	Decision	–	Fail	–	Pass	–	Pass
	99.5893	Decision	–	Fail	–	pass	–	pass

which is the number of pixels changing rate (NPCR), another metric is the unified averaged changed intensity (UACI). It can be defined as the following equations.

$$NPCR = \frac{1}{m \times n} \sum_{i=1}^{m} \sum_{j=1}^{n} \delta(i,j) \times 100\% \tag{35}$$

$$UACI = \frac{1}{m \times n} \left(\sum_{i=1}^{m} \sum_{j=1}^{n} \frac{|C_1(i,j) - C_2(i,j)|}{255} \right) \times 100\% \tag{36}$$

The NPCR takes a high percentage if the positions of the pixels between the two images have been changed. The ideal value for UACI is close to 33% [80]. These metrics are sensitive to any change in images, even if the change is a single-pixel that notices the tiny change. The NPCR calculates and measures the percentage of pixel change in encrypted images and this value means the change in the pixel of the 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 as defined in Eqs. 35 and 36. Table 13 presented all sensitivity metrics for encrypted three images according to three processes of encryption by using six types of PRNGs.

Table 14 Execution time analysis

Images	Encrypted images by using three cipher process																	
	Experimental 1						Experimental 2						Experimental 3					
	Transposition						Substitution						Combination of two methods of encryption					
	MT	dSFMT	MRG	MLFG	shr3cong	SWB	MT	dSFMT	MRG	MLFG	shr3cong	SWB	MT	dSFMT	MRG	MLFG	shr3cong	SWB
Lena	0.297	0.224	0.221	0.207	0.230	0.202	0.2017	0.043	0.023	0.073	0.013	0.043	0.384	0.218	0.239	0.258	0.252	0.200
Baboon	1.656	0.866	0.881	0.909	0.883	0.807	0.114	0.048	0.054	0.055	0.031	0.027	0.866	0.817	0.809	1.053	0.925	0.806
Peppers	1.038	1.381	0.866	1.190	0.881	0.917	0.031	0.033	0.032	0.054	0.027	0.024	1.243	0.821	0.816	0.785	0.840	0.789

Table 15 The NIST measurable test suite

NIST	Key length	PRNG	Permutation key	Decision	XOR key	Decision
Runs test	65,536	MT	0.7354451	Random	0.1279866	Random
		dSFMT	0.8174868	Random	0.5891335	Random
		MRG	0.6944525	Random	0.3494037	Random
		MLFG	0.7170758	Random	0.6610793	Random
		shr3cong	0.5857006	Random	0.2517489	Random
		SWB	0.8766382	Random	0.37527299	Random
Monobit test	65,536	MT	0.9968833	Random	0.9844173	Random
		dSFMT	0.9968833	Random	0.9067115	Random
		MRG	0.9968833	Random	0.9191039	Random
		MLFG	0.9968833	Random	0.9067115	Random
		shr3cong	0.9968833	Random	0.8268448	Random
		SWB	0.9968833	Random	0.9968833	Random
Frequency Test within a Block test	65,536	MT	0	Non-random	0	Non-random
		dSFMT	0	Non-random	0	Non-random
		MRG	0	Non-random	0	Non-random
		MLFG	0	Non-random	0	Non-random
		shr3cong	0	Non-random	0	Non-random
		SWB	0	Non-random	0	Non-random
Longest Run of Ones in a Block test	65,536	MT	0.0200000	Random	0.0200000	Random
		dSFMT	0.0200000	Random	0.0200000	Random
		MRG	0.0200000	Random	0.0200000	Random
		MLFG	0.0200000	Random	0.0200000	Random
		shr3cong	0.0200000	Random	0.0200000	Random
		SWB	0.0200000	Random	0.0200000	Random

4.4.1 Execution time analysis

Good encryption processes depend on the flexibility of the algorithms used to encrypt images and the speed of processes. So, we analyzed and calculated the time required to execute the program for operations. The computational time of the three processes of encryptions among the six types of PRNGs is listed in Table 14. The purpose of the analysis is to confirm the flexible and rapid implementation for each six of the PRNGs. To test the speed of encryption, the test was conducted on 256 and 512 sizes of images and different operations of encryption. Comparing the three experiments, experiment 2 took less time than experiment 1, and experiment 2.

4.4.2 NIST test

The NIST measurable test suite [15] was utilized to assess the randomness of keys. The test includes 15 tests, which detect the appropriation of RNG for randomness in the sequence. These tests were used as the main measure for making randomness decision. In this paper, we work to investigate the randomness by using runs test, Monobit test, Frequency Test within a Block test, and Longest Run of Ones in a Block test. Table 15 shows the outcomes and demonstrates that all $p-$values among 65,536 are uniformly disseminated within the permutation key and XOR key, while the pass rate is additionally satisfactory in three of the NIST test. The standard pass rate is p_value >0.01.

Table 16 Cost effectiveness for each encrypting processes among six types of PRNG

Original Images	PRNG Algo	Encrypted image by three cipher process								
		Experimental 1			Experimental 2			Experimental 3		
		Transposition			Substitution			Combination of two methods		
		(W_1)	(W_2)	Cost	(W_1)	(W_2)	Cost	(W_1)	(W_2)	Cost
Lena 256×256	MT	2.45e+07	2.26e-07	1.09e+14	−9.25e+08	2.18e-09	−4.24e+17	−9.37e+08	4.25e-13	−2.21e+21
	dSFMT	2.57e+07	9.75e-06	2.64e+12	−8.69e+08	−1.86e-09	4.67e+17	−9.68e08	−4.59e-10	2.11e+18
	MRG	2.51e+07	−8.74e-06	−2.87e+12	−1.00e+09	1.90e-09	−5.27e+17	−9.76e+08	1.32e-10	−7.37e+18
	MLFG	2.40e+07	−3.47e-06	−6.90e+12	−9.17e+08	−1.90e-11	4.83e+19	−9.60e+08	−5.72e−10	1.68e+18
	shr3cong	2.41e+07	−1.23e-06	-1.95e+13	−8.72e+08	1.09e-10	−8.03e+18	−9.44e+08	3.13e-12	−3.01e+20
	SWB	2.36e+07	−7.88e-06	−2.99e+12	−8.77e+08	−3.19e-11	2.75e+19	−8.89e+08	5.65e−11	-1.57e+19
Mandrill (Baboon)512×512	MT	2.43e+08	1.73e-05	1.40e+13	7.41e+08	−3.67e-13	−2.02e+21	7.18e+08	6.65e-10	1.08e+18
	dSFMT	2.40e+08	−2.33e-05	−1.03e+13	6.77e+08	−1.68e-12	−4.02e+20	6.83e+08	−1.45e-13	−4.71e+21
	MRG	2.36e+08	−1.15e-05	−2.06e+13	7.16e+08	1.16e-13	6.20e+21	7.01e+08	3.30e-12	2.12e+20
	MLFG	2.38e+08	2.43e-09	9.79e+16	7.3e+08	−8.4e-10	−8.7e+17	6.94e+08	3.83e-10	1.81e+18
	shr3cong	2.38e+08	−7.10e-07	−3.35e+14	7.25e+08	−6.39e-09	−1.13e+17	6.93e+08	−6.05e−10	-1.15e+18
	SWB	2.43e+08	−4.20e-07	−5.78e+14	7.1e+08	9.7e−15	7.3e+22	7.05e+08	3.08e-12	2.29e+20
Peppers512x512	MT	1.81e+08	2.17e-06	8.37e+13	-1.2e+10	2.1e-14	−5.6e+23	−1.16e+10	4.06e-11	−2.85e+20
	dSFMT	1.83e+08	−5.19e-05	−3.52e+12	−1.2e+10	3.99e-15	−2.91e+24	−1.16e+10	−1.11e−11	1.05e+21
	MRG	1.75e+08	−1.02e-07	−1.72e+15	−1.2e+10	1.1e−11	-1.1e+21	-1.16e+10	1.46e-11	−7.94e+20
	MLFG	1.80e+08	−1.03e-05	−1.74e+13	−1.14e+10	−1.9e−12	6.1e+21	-1.16e+10	1.52e-13	−7.61e+22
	shr3cong	1.82e+08	4.56e-07	3.98e+14	−1.15e+10	1.46e-13	−7.90e+22	−1.15e+10	−5.80e-09	1.98e+18
	SWB	1.77e+08	1.74e-04	1.02+12	−1.2e+10	1.5e+10	−7.8e+23	−1.16e+10	7.67e−11	-1.51e+20

5 Research comparison

5.1 Cost-effectiveness

The quality of the encryption system leads to decide the effective algorithm by using several PRNGs. Several statistical parameters can be visualizing the impact of quality in the image encryption system. So, we work on extract the features of each encryption system individually and present a clarification of three existing image encryption systems by using six PRNGs. Due to the extract features and analyze the performances, we achieve one of our goals which are to decide and select the best security PRNGs for processing to enhance and secure PRNGs.

Cost-effectiveness is computed as a figure of merit to combine the comparison parameters similar in principle to the work in reference calculating the weight in paper [5] as the cost-effectiveness equation bellow:

$$Cost = \frac{The\ weight\ of\ \text{the best high values in metrics}}{The\ wight\ of\ \text{the best low values in metrics}} \tag{37}$$

Where, the weight of the best high values in metrics (W_1) is the multiplication of all metrics that achieve high values in image encryption such as entropy, MSE, RMSE, DSSIM, MAE, NAE, SC, MD, AD, AD%, MB, NPCR, UACI, whereas the weight of the best low values in metrics (W_2) is the multiplication of all metrics that achieve low values in image encryption such as PSNR, SNR, IF, SSIM, MS-SSIM, MI, CC, CQ, PCC, H, V, D, ci-square, NCC, time, UIQI.

After applying this computation, the paper presents several remarkable about which system of three encryption methods achieves the high security among six of PRNGs than other rest of the methods. The cost-effectiveness is applying through seven steps. Therefore, we computed the priority weight of several image quality assessments (IQA) with the high and low ideal value in image encryption. The cost-effectiveness results are computed in Table 16 for all six PRNGs experiments in each of the three encryption methods as the following steps.

1. The system computes the cost-effectiveness of each plane of the RGB image separately and then recompiling the image.
2. The system determines the secure type PRNG among the other six types of PRNGs of experiment 1 which uses the permutation method only to encrypt the image.
3. The system determines the secure type PRNG among the other six types of PRNGs of experiment 2 which uses the XOR method only to encrypt the image.
4. The system determines the secure type PRNG among the other six types of PRNGs of experiment 3 which uses the combination methods of permutation and XOR processes to encrypt the image.
5. The system computes the best high values in metrics to get weight in each method (W_1). And select the secure method which has a higher value in quality metrics weight.
6. The system computes the best low values in metrics to get weight in each method (W_2). And select the secure method which has a lower value in quality metrics weight.
7. The cost-effectiveness is computed as the fraction for the higher value in quality metrics weight divided on the lower value in quality metrics weight. Then, the higher value from cost effectiveness decides the secure method among other methods.

A. Comparison of permutation method

Experiment 1 obtained several results of different six permutation keys of PRNGs using for scrambling the images as it is shown in Table 16. Each permutation key is evaluating separately, so Lena's image is encrypted six times individually by using different types of PRNGs and so on for Baboon and Pepper's images.

The effective random generator for using to encrypt Lena's image among other types of PRNGs is Mersenne Twister (MT). MT is selected as an efficient generator because its cost-effectiveness is the highest equals 1.09e+14.

In a similar manner for encrypting the Baboon image, we observed the effective generator by using several performance metrics. The good performance for encrypting the Baboon image is multiplicative lagged Fibonacci (MLFG). Its measuring cost-effectiveness gives the highest value equals 1.40e+13.

Furthermore, we extracted the effective and suitable generator for using in experiment 1 to encrypt the peppers image. Shift-register generator summed with linear congruential generator (shr3cong) achieved the highest cost-effectiveness among the other six PRNGs. The cost-effectiveness equals 3.98e+14.

B. Comparison of the XOR method

Experiment 2 shows all the six keys of PRNGs that are used for encrypting images by using XOR operation. The study runs six different experiments separately by using a single PRNG at each one time from the other six keys of PRNGs. In Table 16 presents the comparison for XORing three images Lena, Baboon, and Peppers.

The effective PRNG using for encrypting Lena image is multiplicative lagged fibonacci (MLFG). The system determining the MLFG as the secure algorithm by computing the cost-effectiveness where it equaled 4.83e+19. So, this experiment considered MLFG with the highest cost.

The efficient way for encrypting Baboon image is XORing image by using modified subtract with borrow generator (SWB) because its cost-effectiveness is the highest equals 7.3e+22 among other generators.

The recommended way for encrypting Peppers image is xoring by using multiplicative lagged fibonacci (MLFG). It achieved the highest cost of equaled 6.1e+21 among the other keys of PRNGs.

Considering the XOR method and depending on higher cost-effectiveness for encrypting the three images, the effective PRNGs are different between Lena, Baboon, and Peppers. It's higher in Baboon and Peppers images than Lena's image. It different because the size is bigger for Baboon and Peppers than Lena. Also, due to the difference in variance for Baboon and Peppers images than Lena's image. The pixels values in Baboon and Peppers images indicate spread more than Lena's image.

C. Comparison combination of two methods

Experiment 3 contains six PRNGs based on the combination of two methods. Which is a permuted block of pixels, and then substituted each pixel value. From this experiment, the system extracted the security method among other keys of PRNGs.

For encrypting Lenas' image, SIMD-oriented Fast Mersenne Twister (dSFMT) is selected as an effective PRNG because its cost is the highest equals 2.11e+18 among other keys of PRNGs. This means the verification of choosing the effectiveness PRNG is the algorithm having the best values in several image quality assessments.

Depending on the cost of effective six of PRNGs, subtract with borrow generator (SWB) seems to be offering the highest cost-effectiveness. Therefore, using the keys of SWB considered a secure algorithm for encrypting the Baboon image because its cost equals 2.29e+20.

For the Peppers image, the outcome demonstrating the highest cost-effectiveness is SIMD-oriented Fast Mersenne Twister (dSFMT). It achieved 1.05e+21 in safety measures strength study. The selection has a higher rate of merit than the other PRNGs.

6 Conclusion

The fundamental point of the paper is to introduce a security system for private image transmission over the web. This paper shall give the experimentation subtleties with the resulted outcomes, indicating which of the PRNG is suitable, a solid and a 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 investigation of six PRNGs by encrypting images with three encryption processes, wherein the first operation of encryption PRNG generates numbers of sequences that are used for shuffling the positions of each pixel of the image to produce an encrypted image with the details being totally wiped. At the second operation of encryption, keys are generated for XOR operations. And the third operation of encryption, examine cipher images by encrypting using a combination of two methods of encryption.

The performance of the PRNG based image encryption methodology was evaluated using several statistical measurements. The experimentations results have shown that the best three PRNGs using in the permutation method are Mersenne Twister (MT), multiplicative lagged Fibonacci (MLFG), and Shift-register generator summed with linear congruential generator (shr3cong) for three encrypted images Lena, Baboon, and Peppers respectively. We obtained the best three PRNGs among substitution cipher experiment, namely multiplicative lagged Fibonacci (MLFG), subtract with borrow generator (SWB), multiplicative lagged fibonacci (MLFG). The observed robust three PRNGs through the third experiment (Combination of permutation and substitution methods) are SIMD-oriented Fast Mersenne Twister (dSFMT), subtract with borrow generator (SWB), and multiplicative lagged fibonacci (MLFG) for three encrypted images Lena, Baboon, and Peppers respectively.

As a future work plan, the continuations of the study of this paper propose to study all cases of the image encryption compared with gray images with the same methodology. This is to show the differences that can occur and study them in terms of changing the data of the encryption method. We can apply all the security measurements studied in this paper to compare the results with them assuming the cost is the same showing possibilities of more attractive remarks.

Acknowledgments This work has been supported by Umm Al-Qura University. We thank our college of computer and information system for providing assistance to make this research possible.

Declarations

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Informed consent Informed consent was obtained from all individual participants included in the study.

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Abbas NA (2016) Image encryption based on independent component analysis and Arnold's cat map. Egypt Inform J 17(1):139–146
2. Ahmad J, Hwang SO (2016) A secure image encryption scheme based on chaotic maps and affine transformation. Multimed Tools Appl 75(21):13951–13976
3. Alharthi N, Gutub A (2017) Data visualization to explore improving decision-making within hajj services. Sci Model Res 2(1):9–18
4. Al-Juaid N, Gutub A (2019) Combining RSA and audio steganography on personal computers for enhancing security. SN Appl Sci 1(8):830
5. AlKhodaidi T, Gutub A (2020) Trustworthy target key alteration helping counting-based secret sharing applicability. Arab J Sci Eng:1–21
6. Al-Najjar Y, Soong D (2012) Comparison of image quality assessment: PSNR, HVS, SSIM, UIQI. Int J Sci Eng Res 3(8):1–5
7. Al-Otaibi N, Gutub A (2014) 2-leyer security system for hiding sensitive text data on personal computers. Lect Notes Inf Theory 2(2):151–157
8. Altalhi S, Gutub A (2021) A survey on predictions of cyber-attacks utilizing real-time twitter tracing recognition. J Ambient Intell Humaniz Comput. https://doi.org/10.1007/s12652-020-02789-z
9. An Enhanced Least Significant Bit Steganography Technique. (2016)
10. Andreatos AS, Leros AP (2014) A comparison of random number sequences for image encryption. Proceedings of MMCTSE, Mathematical Methods & Computational Techniques in Science & Engineering. Athens, Greece, p. 146–151
11. Anley C (2007) Weak randomness: part I—linear congruential random number generators. Next Generation Security Software
12. Ayushi A (2010) Symmetric key cryptographic algorithm. Int J Comput Appl 1(15):1–2
13. Bani MA, Jantan A (2008) Image encryption using block-based transformation algorithm. IJCSNS Int J Comput Sci Netw Secur 8(4):191–197
14. Banthia AK, Tiwari N (2013) Image encryption using Pseudo random number generators. Int J Comput Appl 975:8887
15. Bassham III LE, et al. (2010) Sp 800–22 rev. 1a. a statistical test suite for random and pseudorandom number generators for cryptographic applications. National Institute of Standards & Technology
16. Batool SI, Shah T, Khan M (2014) A color image watermarking scheme based on affine transformation and S 4 permutation. Neural Comput & Applic 25(7–8):2037–2045
17. Behnia S et al (2013) Image encryption based on the Jacobian elliptic maps. J Syst Softw 86(9):2429–2438
18. Bhattacharjee K, Maity K, Das S (2018) A search for good pseudo-random number generators: Survey and empirical studies. arXiv preprint arXiv:1811.04035
19. Chen X, Hu C-J (2017) Medical image encryption based on multiple chaotic mapping and wavelet transform
20. Ding W (2001) Digital image scrambling technology based on Arnold transformation. J Comput Aided Des Comput Graph 13(4):338–341
21. Easttom C (2017) Generating Cryptographic Keys: Will Your Random Number Generators (PRNGs) Do The Job? 22. Available from: https://www.cryptomathic.com/news-events/blog/generating-cryptographic-keys-with-random-number-generators-prng.
22. Elsayed M et al A new method for full reference image blur measure. Int J Simul Syst Sci Technol 19:4
23. Eskicioglu A, Fisher P (1995) Image quality measures and their performance. IEEE Trans Commun 43(12):2959–2965
24. Fathi-Vajargah B, Kanafchian M, Alexandrov V (2018) Image encryption based on permutation and substitution using Clifford chaotic system and logistic map. J Comput 13(3):309–326
25. Francois M et al (2013) A new pseudo-random number generator based on two chaotic maps. Informatica 24(2):181–197

26. Fridrich J (1997) Image encryption based on chaotic maps. In 1997 IEEE international conference on systems, man, and cybernetics. Computational cybernetics and simulation. IEEE
27. Galajda MDP (2006) Chaos-based true random number generator embedded in a mixed-signal reconfigurable hardware. J Electr Eng 57(4):218–225
28. Gao T, Chen Z (2008) Image encryption based on a new total shuffling algorithm. Chaos, Solitons Fractals 38(1):213–220
29. Gashim LL, Hussein KQ (2018) A new algorithm of encryption and decryption of image using combine chaotic mapping. Iraqi J Inf Technol 9(2 16–1 :(انكـ ي ية ا غة ا. https://www.iasj.net/iasj/article/153322
30. Gutub A (2011) Subthreshold sram designs for cryptography security computations. In: International Conference on Software Engineering and Computer Systems. Springer
31. Gutub A, Tenca A (2004) Efficient scalable VLSI architecture for Montgomery inversion in GF (p). Integr VLSI J 37(2):103–120
32. Gutub A, Al-Juaid N, Khan E (2019) Counting-based secret sharing technique for multimedia applications. Multimed Tools Appl 78(5):5591–5619
33. Haahr M (n.d.) Introduction to Randomness and Random Numbers. Available from: https://www.random.org/randomness/
34. Hassan M, Bhagvati C (2012) Structural similarity measure for color images. Int J Comput Appl 43(14):7–12
35. Hussain I, Gondal MA (2014) An extended image encryption using chaotic coupled map and S-box transformation. Nonlinear Dyn 76(2):1355–1363
36. Jagalingam P, Hegde A (2015) A review of quality metrics for fused image. Aquat Procedia 4(Icwrcoe):133–142
37. Janke W (2002) Pseudo random numbers: generation and quality checks. Lect Notes John von Neumann Inst Comput 10:447
38. Kapur V, Paladi ST, Dubbakula N (2015) Two level image encryption using pseudo random number generators. Int J Comput Appl 115:12
39. Kelsey J, et al. (1998) Cryptanalytic attacks on pseudorandom number generators. In International workshop on fast software encryption. Springer
40. Kuo C-J (n.d.) E-mail: jimkuo@ aa. Nctu. Edu. Tw graduate Institute of Communication Engineering National Taiwan University, Taipei, Taiwan, ROC
41. L'Ecuyer P (1999) Good Parameter Sets for Combined Multiple Recursive Random Number Generators II Shorter Version in Operations Research, V. 47I1. P. 159Ä164
42. Li TY, Yorke JA (1975) Period three implies chaos. Am Math Mon 82(1):975
43. Li Y, Wang C, Chen H (2017) A hyper-chaos-based image encryption algorithm using pixel-level permutation and bit-level permutation. Opt Lasers Eng 90:238–246
44. Mao Y, Chen G, Lian S (2004) A novel fast image encryption scheme based on 3D chaotic baker maps. Int J Bifurcation Chaos 14(10):3613–3624
45. Marsaglia G, Tsang WW (2000) The ziggurat method for generating random variables. J Stat Softw 5(8):1–7
46. Marsaglia G, Zaman A (1991) A new class of random number generators. Ann Appl Probab 1(3):462–480
47. Mascagni M, Srinivasan A (2004) Parameterizing parallel multiplicative lagged-Fibonacci generators. Parallel Comput 30(7):899–916
48. Matsumoto M, Nishimura T (1998) Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. ACM Trans Model Comput Simul (TOMACS) 8(1):3–30
49. Mitra A, Rao YS, Prasanna S (2006) A new image encryption approach using combinational permutation techniques. Int J Comput Sci 1(2):127–131
50. Murillo-Escobar M et al (2017) A novel pseudorandom number generator based on pseudorandomly enhanced logistic map. Nonlinear Dyn 87(1):407–425. https://doi.org/10.1007/s11071-016-3051-3
51. Norouzi B et al (2015) A novel image encryption based on row-column, masking and main diffusion processes with hyper chaos. Multimed Tools Appl 74(3):781–811
52. Pak C, Huang L (2017) A new color image encryption using combination of the 1D chaotic map. Signal Process 138:129–137
53. Patidar V, Sud K (2009) A novel pseudo random bit generator based on chaotic standard map and its testing. Electron J Theor Phys 6(20):327–344
54. Peterson G (1997) Arnold's cat map. Math Linear Algebra 45:1–7
55. Petronas U (2011) Mean and standard deviation features of color histogramusing laplacian filter for content-based image retrieval. J Theor Appl Inf Technol 34(1):1–7
56. Preiss J (2015) Color-image quality assessment: from metric to application. PhD Thesis at Technische Universität. https://tuprints.ulb.tudarmstadt.de/4389/1/Preiss_PhD-Thesis.pdf (Accessed June 2021)
57. Pu C (2015) Image scrambling algorithm based on image block and zigzag transformation. Comput Model New Technol:489–493
58. Rajkumar S, Malathi G (2016) A comparative analysis on image quality assessment for real time satellite images. Indian J Sci Technol 9:34

59. Ramasamy P et al (2019) An image encryption scheme based on block scrambling, modified zigzag transformation and key generation using enhanced logistic—tent map. Entropy 21(7):656
60. Ramesh A, Jain A (2015) Hybrid image encryption using Pseudo Random Number Generators, and transposition and substitution techniques. In: 2015 International Conference on Trends in Automation, Communications and Computing Technology (I-TACT-15). IEEE.
61. Rohith S, Bhat KH, Sharma AN (2014) Image encryption and decryption using chaotic key sequence generated by sequence of logistic map and sequence of states of Linear Feedback Shift Register. In: 2014 International Conference on Advances in Electronics Computers and Communications. IEEE
62. Rukhin A, et al. (2001) A statistical test suite for random and pseudorandom number generators for cryptographic applications. Booz-allen and hamilton inc mclean va
63. 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)
64. Saito M, Matsumoto M (2009) A PRNG specialized in double precision floating point numbers using an affine transition, in Monte Carlo and Quasi-Monte Carlo Methods 2008 (p. 589–602). Springer
65. Sang J et al (2018) Joint image compression and encryption using IWT with SPIHT, Kd-tree and chaotic maps. Appl Sci 8(10):1963
66. Saputra I (2017) Image scrambling using one time pad with linear congruent key generator. IJICS Int J Inf Comput Sci **1**(1)
67. Sarma K, Lavanya B (2017) Digital image scrambling based on sequence generation. In: 2017 International Conference on Circuit, Power and Computing Technologies (ICCPCT). IEEE
68. Savvidy KG (2015) The MIXMAX random number generator. Comput Phys Commun 196:161–165
69. Shanon C (1948) A mathematical theory of communication. Bell Syst Tech J 27:379–623
70. Shure SEaL (2001) Matrix Indexing in MATLAB. Available from: https://www.mathworks.com/company/newsletters/articles/matrix-indexing-in-matlab.html
71. Sivakumar T, Devi KG (2017) Image encryption using block permutation and XOR operation. Int J Comput Appl 975:8887
72. Som S, et al. (2014) A chaos based partial image encryption scheme. In: 2014 2nd International Conference on Business and Information Management (ICBIM). IEEE
73. Stallings W (2003) Cryptography and network security: principles and practices,‖ prentice hall. Upper Saddle River
74. Stinson DR, Paterson M (2018) Cryptography: theory and practice. CRC Press.
75. The USC-SIPI Image Database. (n.d.) Available from: http://sipi.usc.edu/database/database.php.
76. Tomassini M, Perrenoud M (2001) Cryptography with cellular automata. Appl Soft Comput 1(2):151–160
77. Al-Roithy B, Gutub A (2020) Trustworthy image security via involving binary and chaotic gravitational searching within PRNG selections. International Journal of Computer Science and Network Security (IJCSNS) 20(12):167–176. https://doi.org/10.22937/IJCSNS.2020.20.12.18
78. Wang Z, Bovik AC (2002) A universal image quality index. IEEE Signal Proc Lett 9(3):81–84
79. Wang Z, Simoncelli EP, Bovik AC (2003) Multiscale structural similarity for image quality assessment. In: The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003. IEEE
80. Wu Y, Noonan J, Agaian S, NPCR and UACI randomness tests for image encryption (2011) Cyber journals: multidisciplinary journals in science and technology. J Sel Areas Telecommun (JSAT) 1(2):31–38
81. Wu Y, Noonan JP, Agaian S (2011) Shannon entropy based randomness measurement and test for image encryption. arXiv preprint arXiv:1103.5520
82. Yang Y-G, Zhao Q-Q (2016) Novel pseudo-random number generator based on quantum random walks. Sci Rep 6(1):1–11
83. Yang Y-G et al (2019) Image compression-encryption scheme based on fractional order hyper-chaotic systems combined with 2D compressed sensing and DNA encoding. Opt Laser Technol 119:105661
84. Ye H-S, Zhou N-R, Gong L-H (2020) Multi-image compression-encryption scheme based on quaternion discrete fractional Hartley transform and improved pixel adaptive diffusion. Signal Process 175:107652
85. Yen K, Yen EK, Johnston RG (1996) The ineffectiveness of the correlation coefficient for image comparisons
86. Younas I, Khan M (2018) A new efficient digital image encryption based on inverse left almost semi group and Lorenz chaotic system. Entropy 20(12):913
87. Yu S-S et al (2020) Optical image encryption algorithm based on phase-truncated short-time fractional Fourier transform and hyper-chaotic system. Opt Lasers Eng 124:105816
88. Zhang L, Tian X, Xia S (2011) A scrambling algorithm of image encryption based on Rubik's cube rotation and logistic sequence. In: 2011 International Conference on Multimedia and Signal Processing. IEEE.
89. Zhang Q, Xue X, Wei X (2012) A novel image encryption algorithm based on DNA subsequence operation. Sci World J 2012
90. Zheng F et al (2008) Pseudo-random sequence generator based on the generalized Henon map. J China Univ Posts Telecommun 15(3):64–68

91. Zhou Q et al (2008) Parallel image encryption algorithm based on discretized chaotic map. Chaos, Solitons Fractals 38(4):1081–1092
92. Zhou NR et al (2015) Quantum image encryption based on generalized Arnold transform and double random-phase encoding. Quantum Inf Process 14(4):1193–1213

Budoor Obid Al-Roithy is currently a graduate student, pursuing Master of Sciences (MS) degree in Computer Sciences & Engineering from Umm Al Qura University (UQU). Her MS program at UQU is specialized in the information security track offered by the College of Computer and Information Systems offered at UQU-Makkah Campus, Saudi Arabia, hoping to complete her research and get the MS degree by 2021.

Adnan Gutub is ranked as Full Professor in Computer Engineering specialized in Information and Computer Security within College of Computers and Information Systems at Umm Al-Qura University (UQU). He has been working as the general supervisor of UQU scientific council following his assignment as Vice Dean of the Custodian of the Two Holy Mosques Institute of the Hajj & Omrah Research, Known publicly as Hajj Research Institute (HRI), within (UQU), Makkah Al-Mukarramah, all Muslims religious Holy City located within the Kingdom of Saudi Arabia. Adnan's academic experience in Computer Engineering was gained from his previous long-time work as Associate Professor, Assistant Professor, Lecturer, and Graduate Assistant, all in Computer Engineering at King Fahd University of Petroleum and Minerals (KFUPM) in Dhahran, Saudi Arabia. He received his Ph.D. degree (2002) in Electrical and Computer Engineering from Oregon State University, USA. He had his BS in Electrical Engineering and MS in Computer Engineering both from KFUPM, Saudi Arabia. Adnan's research work can be observed through his 120+ publications (international journals and conferences) as well as his 5 US patents registered officially by USPTO. His main research interests involved optimizing, modeling, simulating, and synthesizing VLSI hardware for crypto and security computer arithmetic operations. He worked on designing efficient integrated circuits for the Montgomery inverse computation in different finite fields. He has some work in modeling architectures for RSA and elliptic curve crypto operations. His interest in computer security also involved steganography and secret sharing focusing on image-based steganography and Arabic text steganography as well as counting based secret sharing. Administratively, Adnan Gutub filled many executive and managerial academic positions at KFUPM as well as UQU. At KFUPM - Dhahran, he had the experience of chairing the Computer Engineering department (COE) for five years until moving to Makkah in 2010. Then, at UQU - Makkah, Adnan Chaired the Information Systems Department at the College of Computers & Information Systems followed by his leadership of the Center of Research Excellence in Hajj and Omrah (HajjCoRE) serving as HajjCoRE director for around 3-years until the end of 2013. Then, he was assigned his last position (until March 2016) as the Vice Dean of HRI, i.e. the Custodian of the Two Holy Mosques Institute of the Hajj & Omrah Research.