Applied Intelligence (2021) 52:10979–10998
https://doi.org/10.1007/s10489-021-03063-1

Image encryption using permutation generated by modified Regula-Falsi method

Aakash Paul[1] · **Shyamalendu Kandar**[1] · **Bibhas Chandra Dhara**[2]

Accepted: 30 November 2021 / Published online: 18 January 2022

Abstract
Transmission and sharing of multimedia data have drastically increased in the last few years due to the availability of low-cost image capturing devices, development of communication technology, and the popularity of social networks. However several security fissure of a public network such as the Internet has made the jobs of eavesdroppers easy to grab the contents without any impediment. Well-known cipher techniques like DES, AES, RSA can be used to encrypt an image. But, due to the huge volume and high correlation of image data, a lightweight image encryption method is important. Permutation-based encryption methods disrupt the correlation and can act well with huge volumes. Chaos theory is proved to generate pseudo-random sequences and is extensively used to define permutation. A wide range of image encryption proposals based on the permutation defined by the chaotic map is found in the literature. Several non-chaotic techniques are also gaining popularity for defining a permutation. An image encryption proposal based on a non-chaotic method is presented in this current communication. The permutation is defined by the modified Regula-Falsi method and image encryption is achieved by pixel value substitution and iterative addition with the cyclic shift. As the result of the proposed method, fully noisy images are obtained. Security analysis has proved immunity against different attacks. Comparison with state-of-the-art methods has established the applicability of the proposed technique in image encryption.

Keywords Regula-Falsi method · Non chaotic · Image encryption · Iterative addition · Cyclic shift

1 Introduction

Introduction Image transmission through Internet can be found in diverse fields like medical image transmission, sharing personal photographs in social media, confidential military archives, enterprises and storage systems, etc. The tremendous growth in Internet and web technology, availability of image capturing devices, and moreover popularity of social media have provided wings to image transmission. But the transmitted images may be accessed easily by some eavesdroppers through several security fissure of communication channels and those may be exposed to illegal distribution, forgery, etc. To deal with the challenges, different encryption techniques like DES, AES, RSA, IDEA can be thought of for image encryption. These techniques are mainly designed for text data encryption. Since images have bulk amount of data so we need to design a lightweight image encryption technique [1, 2]. Image carries a high correlation among adjacent pixels and scrambling it at pixel or bit-level reduces the correlation and a noise-like image is produced. The scrambling can be performed by a permutation defined from a pseudo-random (PR) sequence. Permutation-based image encryption can handle redundancy, high volume of data and provides speed, less computational overhead than data encryption techniques [3].

Image encryption enters into the research scenario with the SCAN language-based data and image encryption proposal by Bourbakis et al. [4]. The image encryption proposal by Kuo [5] is based on pixel distortion. In earlier image encryption methods, encryption techniques were applied to the compressed image to get faster encryption. Some of the proposals have used quadtree [6], linear quadtree [7],

CTAN.

✉ Shyamalendu Kandar
shyamalenduk@it.iiests.ac.in

[1] Department of Information Technology, Indian Institute of Engineering Science and Technology, Shibpur, India

[2] Department of Information Technology, Jadavpur University, Saltlake Campus, Kolkata, India

Fourier transform [8] etc. to perform image encryption over compressed image.

Image encryption using permutation alters pixel/ bit positions to generate a noise-like image. The permutation must guarantee randomness and high key sensitivity. For this reason, chaos theory has established itself as an alternative choice for image encryption. "The present determines the future, but the approximate present does not approximately determine the future"- is a well-used phrase for chaos theory as a totally different sequence is generated by a little change of the initial parameter/s. Chaos theory has established itself in the area of cryptography with the encryption proposal of Robert Matthews [9]. Permutation of pixel position using chaotic map changes the pixel position and a noise-like image is produced. A number of image encryption techniques [10–14] have employed pixel permutation. Encrypted image generated by pixel permutation has the same histogram as the original and can easily come under the grip of chosen-ciphertext and chosen-plaintext attacks [15–17]. In bit-level permutation (BLP), the image is considered as a binary string and permutation is performed at bit level. This changes the pixels' gray value and histogram attack can be prevented. A number of image encryption proposals [15, 18–25] using bit-level permutation are available in literature. Image as the single binary string is of large size and defining permutation at bit level is time consuming [19]. Several methods have defined bit-level permutation in different ways. Zhu et al. [26] have decomposed the image into eight bitplanes and eight cat maps are used independently to permute the bitplanes. In [27], the plaintext image is decomposed into bitplanes and a bitplane image is taken as a key for XOR operation with the other bitplanes. A scrambling operation of the XORed bitplanes followed by merging to pixel produces an encrypted image. A double cyclic shift is used for bit-level permutation in [23] and it is performed at the pixel level. The amount of cyclic shift is governed by the Henon map. In adaptive image encryption [28], the bitplanes are arranged into two groups, one having bit 1, 2, 7, and 8 and the remaining to another. The hash value of one group is used to permute the bits on the other group in turn using the chaotic map.

In value substitution, a gray value is replaced by another value and it incorporates an extra level of security in image encryption. In [29] pixel value substitution is performed at the time domain where the permutation is defined by the Bernoulli map. 2D sine logistic map is used in [30] to perform value substitution and a faster encrypted image can be obtained. In cryptography, a substitution box, well known as S-Box is a type of symmetric key encryption system to perform the substitution. Adopting substitution mechanism in block cipher obscures the relationship between key used and the ciphertext received. A plenty of proposals [31–37] have adopted S-Box in image encryption.

The development of IoT in the healthcare system requires medical images to be transmitted. The images often used in medical fields; like MRI, Ultrasound sonography, X-ray, etc. contain sensitive medical information which are required to be protected from access by wrong hands while transmitting through a public channel like the Internet. Some recent proposals on medical image encryption are addressed in [38–45].

In recent years DNA encoding has become a prior choice for image encryption. There are four bases A, C, G, and T in a single-strand DNA sequence, where A and T are complements to each other, so are C and G. In binary 00, 01, 10, and 11 are used to represent A, C, G, and T respectively as 00 and 11, 01 and 10 are complementary. From it the number of coding combinations is $4! = 24$. Due to complementary relation, only eight kind of coding combination is achievable. These coding sequences are used to encode 8-bit binary pixels. In recent years, a wide application of DNA encoding and Genetic algorithms are noticed in image encryption proposals [46–50].

Though chaos theory is well accepted in research communities for image encryption; but several non-chaotic methods are also getting popular to generate PR sequence and which are further used in image encryption. Good results are obtained by these non-chaotic techniques, and some of the proposals even give better results than the chaos-based image encryption method. Some of the non chaotic image encryption techniques like Grey code based [51], Rubik's cube principal based [52], prime factorization based [53], circle property based [54], binary tree traversal based [55], cyclic group based [56], interval bisection of polynomial function [57] are available in literature.

It is a proven fact that chaotic maps generate the pseudo-random sequences, thus image encryption based on the permutation defined by pseudo-random sequence generated by chaotic map will return good results. In [54–57] authors have addressed non-chaotic image encryption methods. Using a non-chaotic method the first step is to generate a sequence and to prove it random. This article is pillared on the modified Regula-falsi method to generate the pseudo-random sequence. Generated random sequences have been used in pixel permutation and in pixel value modification. In pixel value modification, the pixel substitution is followed by iterative addition with a cyclic shift operation. High key sensitivity, large keyspace, minimized correlation coefficient, immunity from differential attack, etc. have shown that the proposed technique is robust and secure in comparison to some state-of-the-art image encryption techniques.

The rest of the article is structured as follows. The classical Regula-Falsi method is discussed in Section 2. Permutation generation using modified Regula-Falsi method and the proposed image encryption technique using the permutation is presented in Section 3. Experimental results are

introduced in Section 4. Security analysis of the proposed technique is performed in Section 5. Finally, Section 6 draws the conclusion.

2 Regula-Falsi method

The Regula-Falsi method or false position method is used to find an approximate root of a univariate continuous function $g(x)$. In addition to the function $g(x)$, the method requires two boundary values, a_1 and a_2, such that the condition $g(a_1) \times g(a_2) < 0$ is maintained. In each iteration a straight line S is defined joining the points $(a_1, g(a_1))$ and $(a_2, g(a_2))$. The approximate root x_{new} is the point of intersection of S and the X-axis. If $|g(x_{new})|$ is very close to zero, then x_{new} may be consider as an approximate root. So, if $|g(x_{new})|$ is less than a predefined margin of error τ, the process terminates and x_{new} is returned as the approximation of the root. If not, the process is continued. In that case, if $g(a_1) \times g(x_{new}) < 0$, then x_{new} replaces a_2, otherwise, a_1 is replaced by x_{new} - hence the plausible range where the root lies is narrowed down in each iteration. Algorithm 1 discusses the classical Regula-Falsi method.

Algorithm 1 Falsepos(g(x), a_1, a_2, τ).

Input: $g(x)$: A two or higher degree polynomial. a_1, a_2: Two initial points s.t. $g(a_1) * g(a_2) < 0$, τ: Predefined margin of error
Output: x_{new}: The approximate root
while *true* **do**
  $x_{new} = a_1 - \frac{g(a_1)(a_2 - a_1)}{g(a_2) - g(a_1)}$;
  if $|g(x_{new})| < \tau$ **then**
    break ;
  if $g(a_1) \times g(x_{new}) < 0$ **then**
    $a_2 = x_{new}$;
  else
    $a_1 = x_{new}$;
return x_{new}

The Regula-Falsi method is described with an example. A polynomial $g(x) = 3x^3 - 6x^2 + 7x + 3.2$ with boundary points a_1=-1.5 and a_2=2 (as $g(-1.5) \times g(2)$ = -531.910) and a threshold margin of error τ =0.001 is considered. The newly obtained values in each iteration approaches to the original root as presented in Fig. 1. A straight line joining $(a_1 = -1.5,\ g(a_1) = -30.925)$ and $(a_2 = 2,\ g(a_2) = 17.2)$ is drawn in the first iteration. This is intersected by the X-axis at the point $x_{new} = 0.749$, where the polynomial takes a value of $g(0.749) = 6.337$. Since the points $(-1.5, -30.925)$ and $(0.749, 6.337)$ lie on opposite sides of the X-axis, a_2 takes the value 0.749. Hence the possible range of the root is narrowed down from $(-1.5, 2)$ to $(-1.5, 0.749)$. By this process

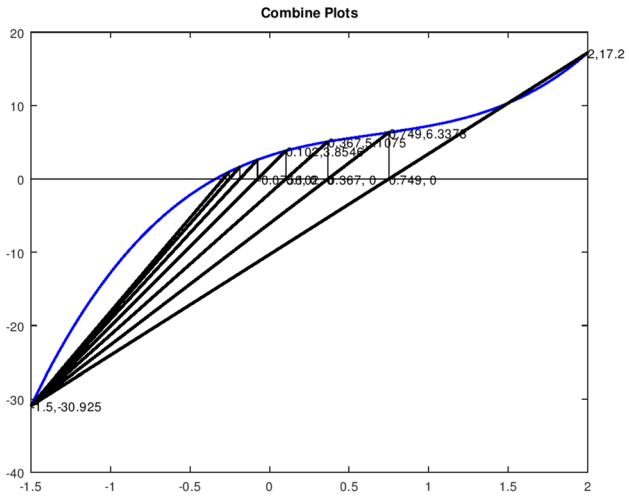


Fig. 1 Plot of six successive steps in Regula-falsi method

in further iterations the plausible range where the root lies is narrowed down. Figure 1 presents four such iterations which clearly signifies that the root is being capsulized within smaller range in each iteration.

Let X_{seq} be the sequence generated by sequentially appending the fraction value of the function $g(x)$ at intersection points x_{new}. In other words, $X_{seq}(i)$ is the value $|g(x_{new})|$ - $\lfloor |g(x_{new})| \rfloor$, received from $g(x)$ at intersection point of the straight line with X-axis, drawn in the i^{th} iteration. Since the range (a_1, a_2) gets narrowed down in each iteration, the sequence X_{seq} asymptotically converges towards the root. Hence, the sequence of intersection points X_{seq} is a convergent sequence, and cannot be considered for long length pseudo random sequence.

3 Proposed method

An image encryption technique based on pseudo-random (PR) sequence generated by the modified Regula-Falsi (MRF) method is addressed in this article. The backbone of the proposed technique is the permutation generated from the PR sequence. The following subsection elaborates the PR sequence generation technique.

The rapid convergence of the Regula-Falsi method towards the root is clearly signified in Fig. 1. The proposed image encryption technique presented in this paper uses a permutation generated by the modified Regula-Falsi method. A PR sequence is required to define the permutation. The various stages of the encryption technique are illustrated in Fig. 2.

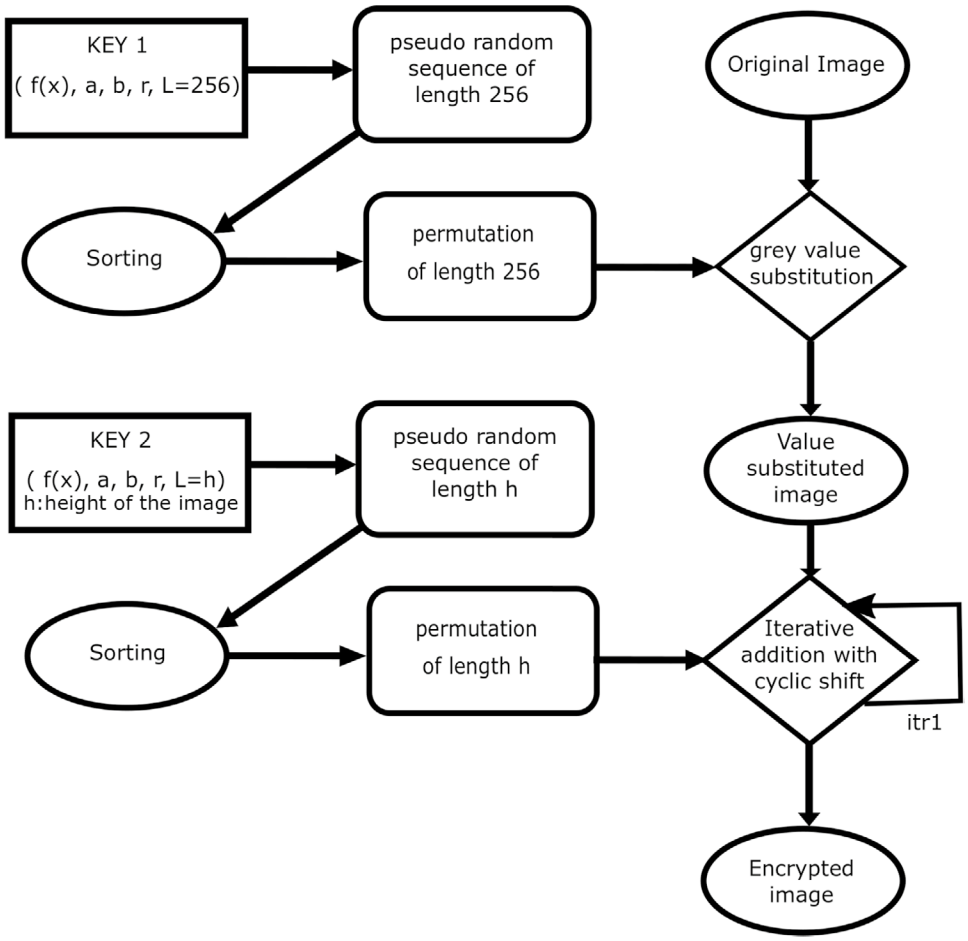


Fig. 2 Block diagram of the proposed technique

3.1 Modified Regula-Falsi method (MRF) and PR sequence generation

In this technique a polynomial $g(x)$ of degree two or more along with two seed points a_1 and a_2 such that $g(a_1) \times g(a_2) < 0$ are taken as input. Other requisite inputs are a real value r and the PR sequence length L. No margin of error τ is considered, to give it the freedom to iterate as many times as necessary, in order to produce a sequence of the desired length.

In this method two values m_1 and m_2 are computed using the fraction part of r. The formula of computation of m_1 and m_2 are swapped in subsequent iteration. Between the points $(a_1, m_1 \times y_1)$ and $(a_2, m_2 \times y_2)$ a straight line S is considered in each iteration (like the classical Regula-Falsi method) where initial values of y_1 and y_2 are $g(a_1)$ and $g(a_2)$, respectively. In each iteration, the straight line S is intersected to the X-axis at some point let $(x_{new}, 0)$. The fraction part of $g(x_{new})$ contributes to the generated sequence. The process is iterated L times to produce sequence of length L. Algorithm 2 illustrates this process.

Algorithm 2 $Prsg_{mrf}(g(x), a_1, a_2, r, L)$.

Input: $g(x)$: A polynomial of degree two or more, a_1 and a_2: seed points such that $g(a_1) \times g(a_2) < 0$, r: a real value, L:sequence length
Output: Seq: PR sequence of length L

$Seq[1, ..., L] = \mathbf{0}$
$y_1 = g(a_1)$
$y_2 = g(a_2)$
$frac = |r| - \lfloor |r| \rfloor$
for $i \leftarrow 1$ *to* L **do**
  if $(i \bmod 2) == 0$ **then**
    $m_1 = (1 - frac)^{-1}$
    $m_2 = (1 + frac)^{-1}$
  else
    $m_1 = (1 + frac)^{-1}$
    $m_2 = (1 - frac)^{-1}$
  $y_1 = m_1 \times y_1$
  $y_2 = m_2 \times y_2$
  $x_{new} = a_1 - \frac{y_1 \times (a_2 - a_1)}{y_2 - y_1}$
  $Seq[i] = |g(x_{new}| - \lfloor |g(x_{new})| \rfloor$
  $frac = Seq[i]$
return Seq

With the help of m_1 and m_2, the value of y_1 and y_2 are changed in each iteration in Algorithm 2. It may be noted

that due to the change in y_1 and y_2 (i.e. $y_1 = m_1 \times y_1$ and $y_2 = m_2 \times y_2$), the points $(a_1,\ y_1)$ and $(a_2,\ y_2)$ move vertically up or down without changing the sign. So, the negativity condition remain same and it ensures the existence of the root within $(a_1,\ y_1)$ and $(a_2,\ y_2)$. The step by step sequence generation process upto 4^{th} iteration using the polynomial $g(x) = 3x^3 - 6x^2 + 7x + 3.2$, with a_1 = -1.5, a_2 = 2 and r=0.75 is presented diagrammatically in Fig. 3. The detailed presentation with the values of x_{new}, $g(x_{new})$ and the terms of *Seq* upto 5^{th} ($L = 5$) iteration are tabulated in Table 1.

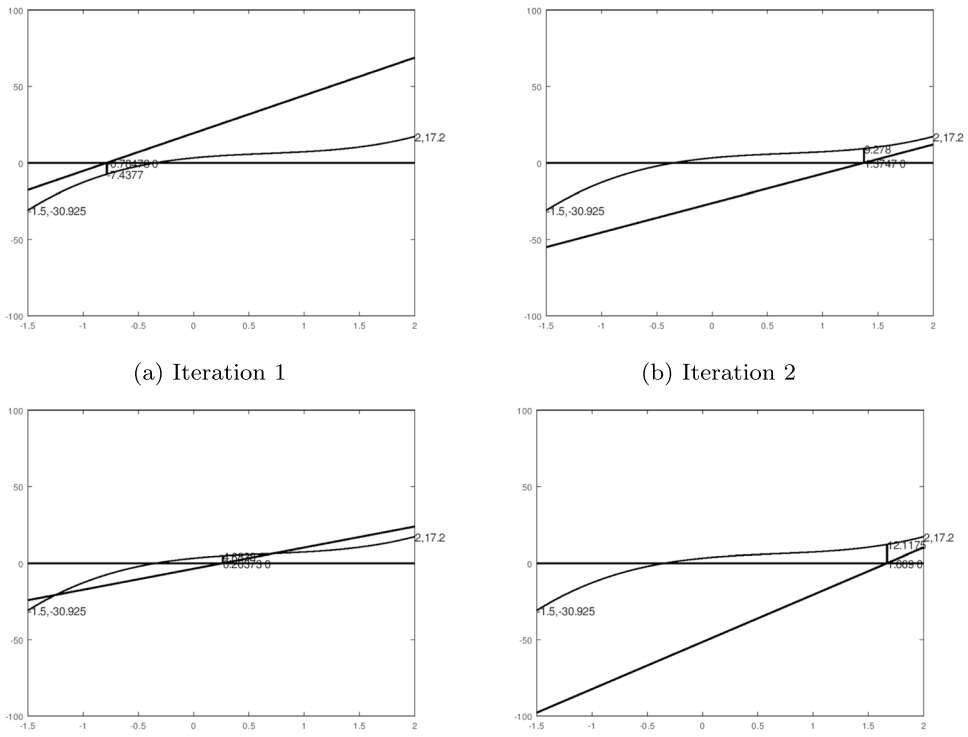
To test the randomness of the MRF method, we consider six polynomials among which two are with degree two, three, and four. The test polynomials with the parameters are given in Table 2. The test polynomials are represented by P_{ij}:$i \in \{2, 3, 4\}$ and $j \in \{1, 2\}$, where P_{ij} is the j^{th} polynomial with degree i. Two types of randomness tests i) NIST randomness test [58] and ii) Dieharder randomness test [59] are performed over the sequence generated by the taken polynomials with respective initial parameters. For each case, 100 binary sequences each of length 10^6 bits are taken.

The results are marked as 'Pass' and 'Fail' (written as 'P' and 'F') for NIST and 'Pass', 'Weak' and 'Fail' (written as 'P', 'W' and 'F') for Dieharder test. The obtained results of both the tests for all the polynomials are presented in Tables 3 and 4 respectively.

It is clear from the results shown in Tables 3 and 4 that the sequences generated by Algorithm 2 have passed the randomness tests for all the polynomials. It is to be noted that the sequences generated by 2^{nd} degree polynomial have failed certain tests of NIST and give 'Weak' results for Dieharder. In the literature, there are many tests to check the existence of certain patterns in the given sequence. These tests are used to comment whether the sequence is non-random or not, i.e., if the test fails (when a certain pattern is detected) then the given sequence is non-random. Even if all the tests are passed then also we cannot make the comment that the given sequence is random. But, it may be expected that the sequence is 'possibly a random' sequence. Using NIST and Dieharder tests (in total 17+30 = 47 tests) we check the existence of certain patterns in the given sequences. But, the sequences pass all the tests (except the sequences generated from 2^{nd} degree polynomials. which are failed to pass certain (two) tests of NIST and Dieharder). We may assume that the sequences, which are generated from the polynomials of degree three or more, are free from the presence of certain patterns. Therefore, we may consider that the sequences generated from the polynomial of degree three or more for further experimentation.

3.2 Encryption technique

The proposed image encryption technique contains two phases a) Substitution Phase b) Iterative Addition with circular shift



(a) Iteration 1 (b) Iteration 2

(c) Iteration 3 (d) Iteration 4

Fig. 3 Successive steps of modified Regula Falsi algorithm upto 4^{th} iteration

Table 1 Intermediate representation of different values in the sequence generation by Algorithm 2 upto first 5 iterations for $3x^3 - 6x^2 + 7x + 3.2$

i	x_{new}	$g(x_{new})$	Seq[i]
1	−0.78473	−7.4377	0.4377
2	1.3747	9.2780	0.2780
3	0.26373	4.6838	0.6838
4	1.6690	12.117	0.1170
5	0.55365	5.7455	0.7455

Table 2 The set of polynomials taken in generating sequences for randomness test

Sl No.	Polynomial	Initial Parameters		
		a	b	r
1.	$2.5x^2 + 5.3x - 4\,(P_{21})$	−2.5	2.5	0.65
2.	$3.1x^2 - x + 2.5\,(P_{22})$	−1	1	0.75
3.	$3x^3 - 6x^2 + 7x + 3.2\,(P_{31})$	−1.5	2	0.75
4.	$2.5x^3 - 4.2x^2 - 3.7x + 2.8\,(P_{32})$	−1.42	1.83	0.72
5.	$1.7x^4 - 2.1x^3 - 1.3x^2 + 2.2x - 1.9\,(P_{41})$	−1.4	1.7	0.69
6.	$x^4 - 5x^3 + 5x^2 + x - 6\,(P_{42})$	−1.5	4	0.73

The backbone of the proposed encryption technique is the pseudo-random sequence received from Algorithm 2.

In the first phase, permutation *Per* is generated from the PR sequence produced by Algorithm 2. The permutation is used to substitute the gray value of a pixel with another gray value. In the second phase, iterative addition with circular shift makes the image fully encrypted. Figure 2 presents the block diagram of the proposed image encryption technique.

3.2.1 Substitution phase

The gray value of a pixel is replaced by another gray value in the substitution phase. This can be achieved by permutation *Per*. The construction of *Per* starts with generating a sequence *Seq* of length(L) 256 (=2^8 for 8 bit grayscale image) using Algorithm 2. The permutation *Per* is received from the sequence *Seq* by sorting.

In the actual substitution phase, each pixel of the image *Img* is replaced by a value. A substitution index *index* and the permutation *Per* are computed for each location from the pixel of the original image. Each pixel of the original image is substituted by *Per*[*index*] to produce the substituted image Img_{sub}. Algorithm 3 describes the substitution phase.

Algorithm 3 Substitution($Img, g(x), a_1, a_2, r, L$).

Input: Original image $Img_{h\times w}$, $g(x)$, a_1, a_2 and r are for input in Algorithm 2, Length of the sequence $L = 256$.
Output: substituted image Img_{sub}
$Seq = prsg_{mrf}(g(x), a_1, a_2, r, 256)$
for $i \leftarrow 1$ *to* 256 **do**
  $tup(i) = (seq[i], i)$
$tup_{sort} = sort(tup)$
for $i \leftarrow 1$ *to* 256 **do**
  $item = tup_{sort}[i]$ // item take a tuple
  $Per[i] = item(2)^{\dagger}$ // 2^{nd} value of tuple
for $i \leftarrow 1$ *to* h **do**
  for $j \leftarrow 1$ *to* w **do**
    $index = (Img(i, j) + Per((i \bmod 256) + 1)) \bmod 256 + 1$
    $Img_{sub}(i, j) = Per(index)$
return Img_{sub}
† item(2) contains the index positions of the sorted tup

3.2.2 Iterative addition with cyclic shift

In the substitution phase, for a given row i, all occurrences of pixel value p shall be replaced by a particular gray value q. However, for some other row l, such that $l \bmod 256 \neq i \bmod 256$, the value p shall be replaced by a value r, different from q. To achieve more security, iterative addition with cyclic shift phase is performed.

The iterated addition with cyclic shift phase is an iterative process, carried out in row-wise fashion on image I. Here a permutation *Per* of length h is defined from the sequence generated in Algorithm 2. The pixels of substituted image $Img_{sub}(i,j)$ is defined to $Img_{acs}(i,j)$ with the help of *Per* and pixel of Img_{acs} computed one step earlier. Iterative addition with cyclic shift is iterated a number of times let itr_1 to make it sensitive to any change (Avalanche effect). Iterative addition with cyclic shift operation is presented in Algorithm 4.

Algorithm 4 Iacs($I_{w\times h}, g(x), a_1, a_2, r, L$).

Input: $I_{w\times h}$: Input Image
$g(x)$, a_1, a_2 and r are for input in Algorithm 2, length $L = h$
Output: encrypted Image Img_{acs} size $h * w$
$Seq = Prsg_{mrf}(g(x), a_1, a_2, r, h)$
Per = permutation acheived by sorting Seq
$b = I[Per(h)][w]$
for $i \leftarrow 1$ *to* h **do**
  $rw = Per(i)$
  for $j \leftarrow 1$ *to* w **do**
    $x = (I[rw][j] + b) \bmod 256$
    $Img_{acs}[rw][j] = BCS(x, (Per(j) \bmod 8))^{\dagger}$
    $b := Img_{acs}[rw][j]$
return Img_{acs}
†BCS: Bitwise Circular Shift

Table 3 NIST Randomness Test results of MRF method over different polynomials

	P_{21}		P_{22}		P_{31}		P_{32}		P_{41}		P_{42}	
Test Name	p value	Comment	p value	Comment	value	Comment	p value	Comment	p value	Comment	p value	Comment
Frequency	0.474986	P	0.419021	P	0.275709	P	0.315206	P	0.503135	P	0.340895	P
Runs	0.025193	P	0.085587	P	0.437274	P	0.193808	P	0.898736	P	0.233025	P
Cumulative Sums (Forward)	0.455937	P	0.080519	P	0.437274	P	0.692143	P	0.236819	P	0.462931	P
Cumulative Sums (Backward)	0.678686	P	0.122325	P	0.699313	P	0.220438	P	0.540413	P	0.362941	P
Block Frequency	0.334538	P	0.419021	P	0.249284	P	0.384202	P	0.318354	P	0.205373	P
Longest Runs	0.911413	P	0.719747	P	0.574903	P	0.430116	P	0.656896	P	0.928136	P
Rank	0.275709	P	0.971699	P	0.319084	P	0.917207	P	0.336187	P	0.536578	P
Random Execution	0.437274	P	0.275709	P	0.041438	P	0.189164	P	0.578120	P	0563860	P
FFT	0.030806	P	0.383827	P	0.834308	P	0.041613	P	0.489765	P	0.881905	P
Non Overlapping Template	0.816537	P	0.474986	P	0.935716	P	0.837419	P	0.859758	P	0.749866	P
Overlapping Template	0.851383	P	0.102374	P	0.437274	P	0.284935	P	0.176463	P	0.487931	P
Linear Complexity	0.867692	P	0.964295	P	0.514124	P	0.643840	P	0.578129	P	0.875900	P
Universal	0.534146	P	0.275709	P	0.102526	P	0.221037	P	0.426810	P	0.470321	P
Serial (Forward)	0.000010	F	0.102526	P	0.162606	P	0.051983	P	0.121086	P	0.085108	P
Serial (Backward)	0.000000	F	0.085587	P	0.6999313	P	0.097801	P	0.163561	P	0.586428	P
Approximate Entropy	0.231050	P	0.162606	P	0.213309	P	0.165730	P	0.218625	P	0.174218	P
Random Execution Variant	0.178272	P	0.110952	P	0.392456	P	0.183931	P	0.344118	P	0.102850	P

'P' denotes Pass and 'F' denotes Fail

Table 4 Dieharder Randomness Test results of MRF method over different polynomials

Sl.	Test Name	P_{21}		P_{22}		P_{31}		P_{32}		P_{41}		P_{42}	
		p value	Comment	p value	Comment	p value	Comment	p value	Comment	p value	Comment	p value	Comment
1	Diehard Birthdays Test	0.56522933	P	0.3972191	P	0.7315102	P	0.33028368	P	0.79321459	P	0.26917053	P
2	Diehard OPERM5 Test	0.35280334	P	0.75818064	P	0.87436841	P	0.71974064	P	0.92713082	P	0.03653943	P
3	Diehard 32x32 Binary Rank Test	0.79466004	P	0.61351968	P	0.61247694	P	0.65945421	P	0.91884499	P	0.83561317	P
4	Diehard 6x8 Binary Rank Test	0.22607877	P	0.46846657	P	0.76812677	P	0.6864359	P	0.29985309	P	0.72283213	P
5	Diehard Bitstream Test	0.97612927	P	0.31578398	P	0.93514441	P	0.30556904	P	0.07945656	P	0.59131038	P
6	Diehard OPSO	0.04845962	P	0.98417331	P	0.76698845	P	0.79364186	P	0.53238383	P	0.60281114	P
7	Diehard OQSO Test	0.82236065	P	0.7772127	P	0.37645826	P	0.27101361	P	0.73411505	P	0.83061517	P
8	Diehard DNA Test	0.40347209	P	0.26119279	P	0.2183787	P	0.03677633	P	0.40562611	P	0.06271383	P
9	Diehard Count the 1s (stream) Test	0.43581177	P	0.43070553	P	0.10755154	P	0.54117509	P	0.16974861	P	0.96570438	P
10	Diehard Count the 1s Test (byte)	0.97131342	P	0.4915993	P	0.45335651	P	0.93317977	P	0.98058912	P	0.16698847	P
11	Diehard Parking Lot Test	0.99951901	W	0.24635381	P	0.87504329	P	0.98287267	P	0.43824831	P	0.43004782	P
12	Diehard 2dsphereTest	0.57524854	P	0.64292283	P	0.26112755	P	0.31205172	P	0.43602699	P	0.6143719	P
13	Diehard 3d Sphere (Minimum Distance) Test	0.74978622	P	0.66199534	P	0.63153241	P	0.74046438	P	0.67398086	P	0.39149727	P
14	Diehard Squeeze Test	0.45927607	P	0.34539554	P	0.89924702	P	0.53513192	P	0.44004981	P	0.51024546	P
15	Diehard Sums Test	0.51189233	P	0.02759492	P	0.04962597	P	0.03952903	P	0.01227701	P	0.43822401	P
16	Diehard Runs Test	0.48725533	P	0.11050108	P	0.87506427	P	0.26156154	P	0.4929924	P	0.59033144	P
17	Diehard Craps Test	0.89719465	P	0.40921112	P	0.93788149	P	0.64108356	P	0.52038969	P	0.71440275	P
18	Marsaglia and Tsang GCD Test	0.02426115	P	0.5092093	P	0.0690417	P	0.56011523	P	0.52611263	P	0.22852138	P
19	STS Monobit Test	0.57012144	P	0.42194466	P	0.66095606	P	0.60089132	P	0.08712629	P	0.69434884	P
20	STS Runs Test	0.16403334	P	0.75747591	P	0.45548713	P	0.68551337	P	0.88176079	P	0.38089727	P
21	STS Serial Test (Generalized)	0.6379042	P	0.2841831	P	0.74957928	P	0.74332333	P	0.31367234	P	0.47754612	P
22	RGB Bit Distribution Test	0.47829125	P	0.43775893	P	0.77095666	P	0.44896759	P	0.85463355	P	0.62473294	P
23	RGB Generalized Minimum Distance Test	0.00139914	W	0.17719732	P	0.73044154	P	0.15309589	P	0.83936342	P	0.34885075	P
24	RGB Permutations Test	0.48099984	P	0.22415869	P	0.23914867	P	0.05401112	P	0.81757203	P	0.59142556	P
25	RGB Lagged Sum Test	0.62549315	P	0.32202137	P	0.62314073	P	0.09544068	P	0.25367132	P	0.48100584	P
26	RGB Kolmogorov-Smirnov Test Test	0.15957916	P	0.18053026	P	0.54315297	P	0.24604393	P	0.35887654	P	0.93041102	P
27	dab_bytedistrib	0.60233608	P	0.0019609	W	0.22732672	P	0.21896712	P	0.8606419	P	0.1598194	P
28	dab_dct	0.73352119	P	0.57133575	P	0.47285837	P	0.81788767	P	0.88361951	P	0.93028196	P
29	dab_filltree	0.59830347	P	0.67415319	P	0.03196556	P	0.57295739	P	0.25160123	P	0.03121103	P
30	dab_monobit2	0.72468563	P	0.23674473	P	0.90291475	P	0.21824167	P	0.3436104	P	0.50042792	P

'P' denotes Pass and 'W' denotes Weak

4 Experimental result

Twelve 8 bit grayscale images of size 512×512 are considered in this experiment. There are 8 standard test images 'Baboon', 'Barbara', 'Cameraman', 'House', 'Jetplane', 'Lake', 'Lena', 'Pepper'. The test set also contains two special grayscale images, one is 'Checkerboard' (contains 50% white and 50% black pixels) and a constant image 'Black' (having only black color). To have some test results over medical images we have included 'Chest X-RAY' and 'Brain MRI' into the image set. The test images are shown in Fig. 4.

For the comparison purpose of the proposed technique with some recent proposals we have considered four techniques namely Diaconu et al. [20], Zhang et al. [31], Kumar et al. [60] and Kandar et al. [56] scheme. The first three techniques are chaos-based methods whereas the last one is a non-chaotic technique.

The proposed encryption method consists of two techniques 'substitution' and 'iterrative addition with cyclic shift' and for that two pseudo-random sequences of length 256 and h (the height of the original image) respectively are required to be produced using Algorithm 2. Two separate key sets in form of $\{g(x), a_1, a_2, r, itr_1\}$ are needed to generate those sequences. In this experiment, we have produced a sequence of length $(256 + h)$ from the same set of keys for simplicity. As mentioned earlier, for the experimentation purpose we have taken polynomials of degree three and four. The polynomials with initial parameters are given in Table 2.

The experimental results over eight images namely 'Baboon', 'Cameraman', 'Lena', 'Peppers', 'Black', 'Checkerboard', 'Chest X-RAY' and 'Brain MRI' are presented in Fig. 5. It is to be noted that the substituted images (received from Algorithm 3) and final encrypted images (received from Algorithm 4) are presented under headings 'Substituted' and 'Encrypted' in Fig. 5. From the experimental results, it is found that there is some hazy row-wise line in the substituted images (mainly in Cameraman, Black, Checkerboard, Chest X-RAY, and Brain MRI). These are coming due to row-wise replacement of the same gray value by some other gray value. Whereas fully noisy images are noticed for the final encrypted images from which no visual information about the original image can be received. For the case of two medical images, noise-like images are received in encrypted versions. Even the black and checkerboard images are received as noisy in the encrypted version. From the results, the proposed image encryption technique can be marked as a good one.

To compute the execution time, we consider images with different sizes (128×128, 256×256, 512×512 and 1024×1024), derived from the test images by downsampling and upsampling (repetition of rows and columns). The test was performed in a 2.3 GHz Intel core i5 processor system, using MATLAB R2018b. The average execution times of the encryption technique for each category grouped by size are reported in Table 5. The proposed method is fast in execution as reflected by the table.

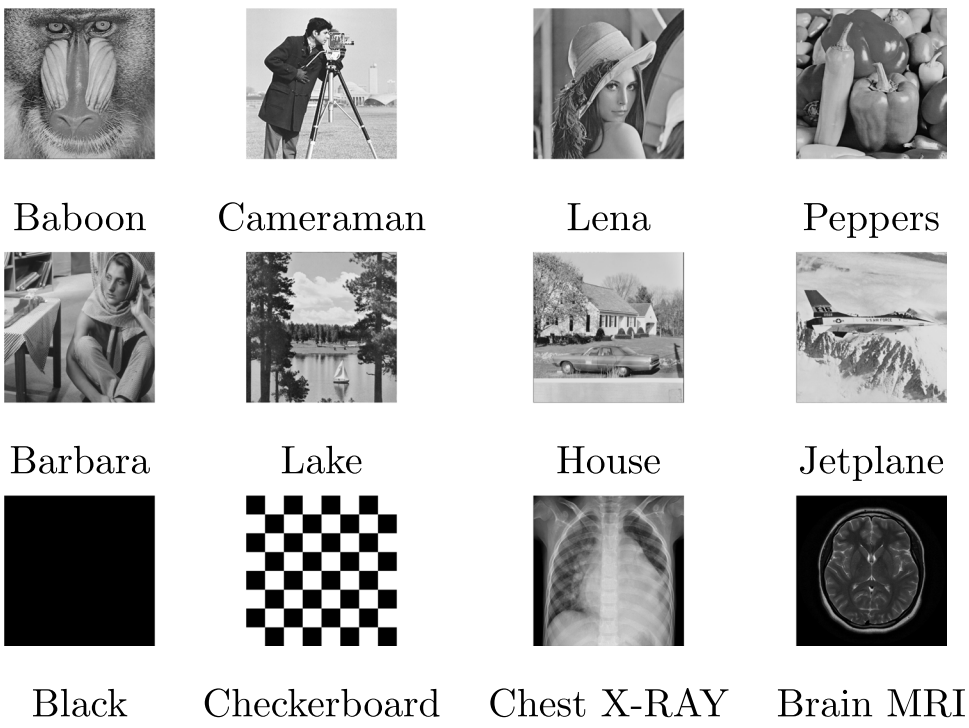


Fig. 4 The test images for experimental purpose

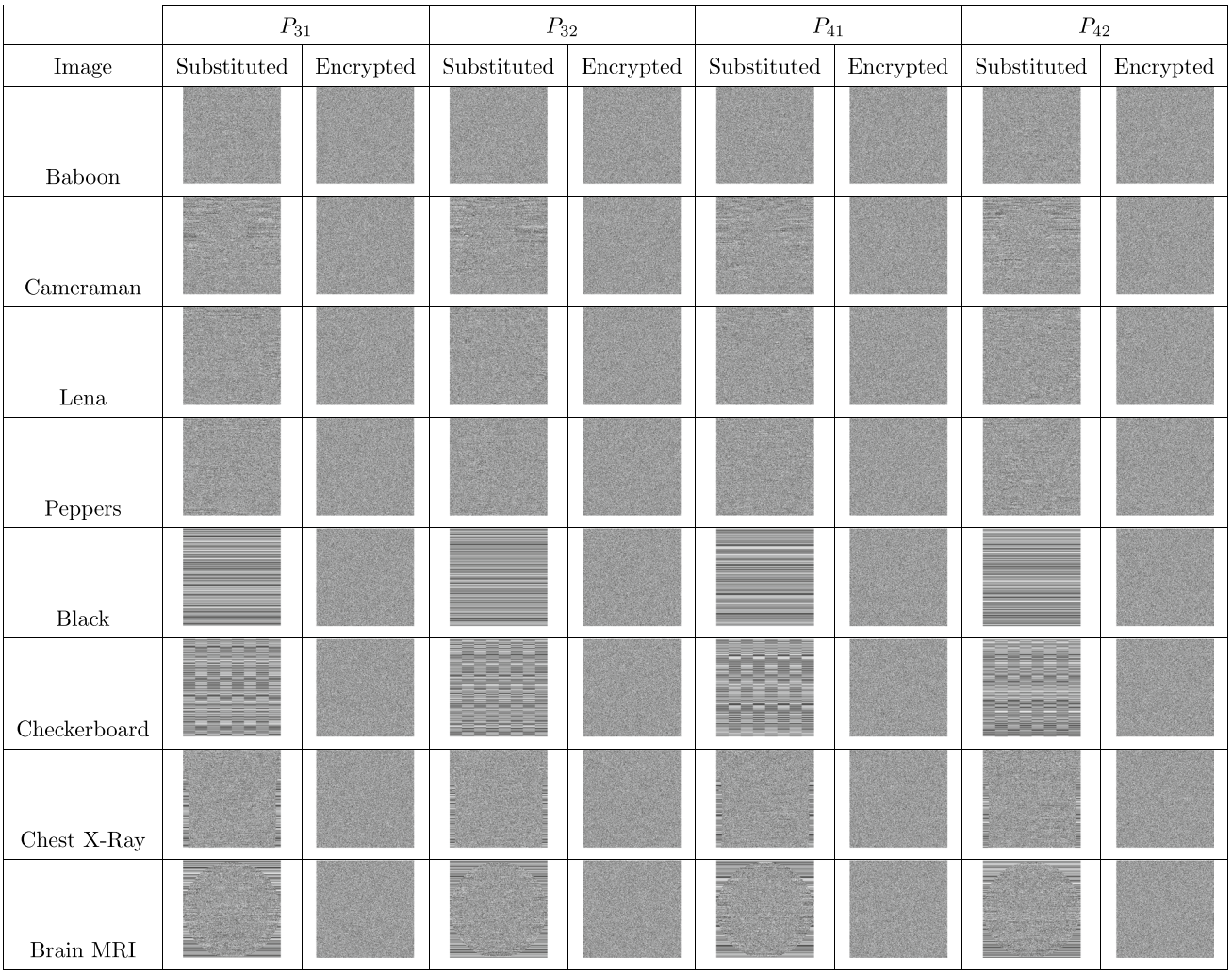


Fig. 5 Substituted and Encrypted images

5 Security analysis

There must exist some degree of robustness of an encryption technique so that it becomes really difficult to an attacker to recover sensitive information related to the plaintext from the encrypted one. Some standard matrices and statistical tests described in the following subsection are applied to measure the robustness and security of the image encryption techniques.

5.1 Secret key space

Keys play a trivial part in PR sequence generation. PR sequence can be generated easily if the keys are compromised somehow. This results in the retrieval of the original image. Limited key space makes the process easily vulnerable to the attacker using brute force attack, to prevent which a good encryption technique must have a sufficiently large keyspace.

Table 5 Average encryption time of images of different sizes

Size	Average Execution Time(in seconds)
128×128	0.085890
256×256	0.256722
512×512	0.620413
1024×1024	2.895086

For the proposed technique; the polynomial function $g(x)$, two seed points a_1, a_2 and a real value r define the key space. Number of iteration itr_1 of iterative addition with cyclic shift is also considered as a key. Two separate polynomials can be used in generating sequences for substitution and iterative addition with cyclic shift phase. As a simplified approach we have maintained $g(x)$, a_1, a_2 and r same to produce all the PR

sequences required. A polynomial $g(x)$ can be represented in two ways. i) By (cof_i, exp_i) for $1 \leq i \leq k$ where cof denotes coefficient, exp denotes exponent and k is the number of terms or by ii) $(m+1)$ coefficients where m denotes the degree of the polynomial. If each term is represented by 64 bit floating point, then the key space will be $(2k+3) \times 64 + 3$ bits or $(m+1+3) \times 64 + 3$ bits (inner '+3' is for 'a_1','a_2' and 'r'. Outer '+3' is for itr_1). The large key space makes it tough for an attacker to correctly form the polynomial $g(x)$ and the seed values using brute force attack.

5.2 Information entropy analysis

Information entropy E of a message M is computed as

$$E(M) = \frac{1}{N} \sum_{i=1}^{N} p(M_i) log_2 \frac{1}{p(M_i)} \quad (1)$$

Here the probability of symbol the M_i contains in the message M is denoted by $p(M_i)$. N represents a total number of symbols. Maximum attainable entropy is 8 of an 8-bit grayscale image. The maximum entropy value denotes each gray value with equal probability. Whereas, a certain degree of exposer to the original is signified by an entropy value less than 8.

Table 6 presents the information entropy of the original and encrypted images. The table reflects that for all test images the information entropy is very near to 8 in their encrypted version. This signifies uncertainty of the proposed technique in an unknown platform, so it can be marked as secure.

The proposed method is compared with the four earlier mentioned techniques in terms of information entropy and the results are listed in Table 7. It is noted that the average results for 'Lena', 'Baboon' and 'Peppers' are considered for the comparison. It is observed that the performance of the proposed technique is as close to the state-of-the-art methods.

5.3 Key sensitivity analysis

The key sensitivity of an image encryption technique can be measured in two ways - from the encryption point and from the decryption point.

i) From an encryption point of view, image encryption is marked as sensitive if two completely different encrypted images are produced with a slide change of the encryption key.
ii) From the decryption point of view, an image encryption technique is marked as sensitive if a noise-like image is produced, in choosing a slightly changed decryption key.

Key sensitivity is measured by the percentage of changed pixels in the two encrypted or the two decrypted images. In the proposed method the keys are the seed points a_1 and a_2, a real value r and number of iteration itr_1. For key sensitivity analysis, the change is made only to a_1 and r but the polynomial $g(x)$ and number of iterations itr_1 are kept unaltered for simplicity. The test configuration for key sensitivity analysis of the technique is presented in Table 8

I. (i) 'Lena' image is encrypted to I' and I'' using polynomial $3x^3 - 6x^2 + 7x + 3.2$ with keys $(-1.5,\ 2,\ 0.75)$ and $(\mathbf{-1.5000001},\ 2,\ 0.75)$ respectively. It is found that 99.598694% pixels are different while comparing I and I''. Two encrypted images, the difference image

Table 6 Information Entropy of original images and their encrypted versions

Sl	Image	Information Entropy				
		Original	Encrypted			
			P_{31}	P_{32}	P_{41}	P_{42}
1	Baboon	7.3579490	7.9991714	7.9992632	7.9993960	7.9993939
2	Barbara	73537898	7.9991598	7.9993501	7.9993525	7.9992614
3	Cameraman	6.9565594	7.9991445	7.9994033	7.9993283	7.9992810
4	House	7.6547525	7.9993092	7.9992453	7.9991877	7.9992125
5	Jetplane	6.7024627	7.9993323	7.9993440	7.9993182	7.9993228
6	Lake	7.4842192	7.9992362	7.9992356	7.9992414	7.9992405
7	Lena	7.4455067	7.9993002	7.9992108	7.9991705	7.9993983
8	Peppers	7.5936546	7.9993978	7.9992526	7.9993206	7.9993328
9	Black	0	7.9992350	7.9993080	7.9992849	7.9991956
10	Checkerboard	1	7.9993234	7.9994070	7.9993836	7.9992347
11	Chest X-RAY	7.3403152	7.9993376	7.9993443	7.9993376	7.9993607
12	Brain MRI	5.3969247	7.9993499	7.9992491	7.9992621	7.9992331

Table 7 Information Entropy analysis

	State-of-the-art methods				Proposed Method
	Diaconu et al. [20]	Zhang et al. [31]	Kumar et al. [60]	Kandar et al. [56]	
Lena	7.998025	7.999324	7.999600	7.999360	7.999269
Baboon	7.997124	7.999285	7.999200	7.999348	7.999306
Peppers	7.998521	7.999314	7.999300	7.999354	7.999326
Average	7.997890	7.999307	7.999366	7.999321	7.999300

$(I'' - I')$ and its corresponding histogram are shown in Fig. 6b to e.

II. (ii) 'Lena' image is encrypted to I' and I'' using polynomial $3x^3 - 6x^2 + 7x + 3.2$ with keys $(-1.5,\ 2,\ 0.75)$ and $(-1.5,\ 2,\ \mathbf{0.75000001})$ respectively. It is found that 99.608612% pixels are different while comparing I and I''. Two encrypted images, the difference image $(I'' - I')$ and its corresponding histogram are shown in Fig. 6b, f to h.

III. (i) 'Lena' image (I) is encrypted to I' using polynomial $3x^3 - 6x^2 + 7x + 3.2$ with keys $(-1.5,\ 2,\ 0.75)$. I' is decrypted to I'' using keys $(\mathbf{-1.5000001},\ 2,\ 0.75)$. It is found that 99.605179% pixels are different while comparing 'Lena' and I''. Figure 6a, b and i present respectively the original, encrypted and decrypted image (using slidely different key) .

IV. (ii) 'Lena' image (I) is encrypted to I' using polynomial $3x^3 - 6x^2 + 7x + 3.2$ with keys $(-1.5,\ 2,\ 0.75)$. I' is decrypted to I'' using keys $(-1.5,\ 2,\ \mathbf{0.7500001})$. It is found that 99.621963% pixels are different while comparing 'Lena' and I''. Figure 6a, b and (j) present respectively the original image, encrypted image and decrypted image (using slidely different key).

The results for the key sensitivity of the image encryption technique for all the polynomials of degree 3 and degree 4 are listed in Table 9. It can be remarked from the obtained results that the proposed method is sensitive to any change of the key.

5.4 Statistical attack

Digital image contains a number of correlated pixels. The statistical weakness of a cryptosystem is the key target of an attacker in a statistical attack. In image encryption, mainly two types of statistical attack are the point of a target for an attacker.

i) Histogram attack, where the difference of the original image histogram and the encrypted image histogram is observed and ii) correlation coefficient analysis, where correlation coefficient of original and encrypted images are analyzed.

5.5 Histogram attack

The histogram represents the frequency of occurrence of distinct pixel intensity values of a digital image graphically. Some statistical information - mainly the tonal distribution of the original image may be revealed from the histogram. A histogram having equal or least variation of the frequency of pixels does not disclose useful information about the original image. Thus a flat or nearly flat histogram is desirable for a good image encryption technique. The histograms of the original and final encrypted images of 'Lena', 'Baboon', 'Cameraman', 'Peppers', 'Checkerboard', 'Black', 'Chest X-RAY' and 'Brain MRI' are presented in Fig. 7. It is observed from Fig. 7 that the encrypted images' histograms are nearly unvarying and are different from that of the original versions. It can be remarked from

Table 8 Test configuration for key sensitivity analysis

	(i)	(ii)
I. Encryption key is changed	I is encrypted to I' by (a_1, a_2, r)	I is encrypted to I' by (a_1, a_2, r)
	I is encrypted to I'' by (a_1', a_2, r)	I is encrypted to I'' by (a_1, a_2, r')
	I' and I'' are compared	I and I'' are compared
II. Decryption key is changed	I is encrypted to I' by (a_1, a_2, r)	I is encrypted to I' by (a_1, a_2, r)
	I' is decrypted to I'' by (a_1', a_2, r)	I' is decrypted to I'' by (a_1, a_2, r')
	I and I'' are compared	I and I'' are compared

Table 9 Key sensitivity of 'Lena' image for degree 3 and degree 4 polynomials

		(i)	(ii)
I.	P_{31}	'Lena' is encrypted to I' by (−1.5, 2, 0.75) 'Lena' is encrypted to I'' by (−1.5000001, 2, 0.75) I' and I'' are **99.598694**% different	'Lena' is encrypted to I' by (−1.5, 2, 0.75) 'Lena' is encrypted to I'' by (−1.5, 2, **0**.75000001) I' and I'' are **99.608612**% different
	P_{32}	'Lena' is encrypted to I' by (−1.42, 1.83, 0.72) 'Lena' is encrypted to I'' by (−1.4200001, 1.83, 0.72) I' and I'' are **99.623108**% different	'Lena' is encrypted to I' by (−1.42, 1.83, 0.72) 'Lena' is encrypted to I'' by (−1.42, 1.83, **0**.72000001) I' and I'' are **99.601746**% different
II.	P_{31}	'Lena'(I) is encrypted to I' by (−1.5, 2, 0.75) I' is decrypted to I'' by (−1.5000001, 2, 0.75) I and I'' are **99.605179**% different	'Lena'(I) is encrypted to I' by (−1.5, 2, 0.75) I' is decrypted to I'' by (−1.5, 2, **0**.7500001) I and I'' are **99.621963**% different
	P_{32}	'Lena'(I) is encrypted to I' by (−1.42, 1.83, 0.72) I' is decrypted to I'' by (−1.4200001, 1.83, 0.72) I and I'' are **99.614788**% different	'Lena'(I) is encrypted to I' by (−1.42, 1.83, 0.72) I' is decrypted to I'' by (−1.42, 1.83, **0**.7200001) I and I'' are **99.605832**% different
I.	P_{41}	'Lena' is encrypted to I' by (-1.4, 1.7, 0.69) 'Lena' is encrypted to I'' by (**−1.4000001**, 1.7, 0.69) I' and I'' are **99.612808**% different	'Lena' is encrypted to I' by (-1.4, 1.7, 0.69) 'Lena' is encrypted to I'' by (-1.4, 1.7, **0.6900001**) I' and I'' are **99.617385**% different
	P_{42}	'Lena' is encrypted to I' by (-1.5, 4, 0.73) 'Lena' is encrypted to I'' by (−1.5000001, 4, 0.73) I' and I'' are **99.626922**% different	'Lena' is encrypted to I' by (-1.5, 4, 0.73) 'Lena' is encrypted to I'' by (-1.5, 4, **0**.7300001) I' and I'' are **99.607086**% different
II.	P_{41}	'Lena'(I) is encrypted to I' by (-1.4, 1.7, 0.69) I' is decrypted to I'' by (**-1.5000001**, 4, 0.73) I and I'' are **99.619740**% different	'Lena'(I) is encrypted to I' by (-1.4, 1.7, 0.69) I' is decrypted to I'' by (-1.4, 1.7, 0.6900001) I and I'' are **99.610483**% different
	P_{42}	'Lena'(I) is encrypted to I' by (-1.5, 4, 0.73) I' is encrypted to I'' by (**-1.5000001**, 4, 0.73) I' and I'' are **99.626922**% different	'Lena'(I) is encrypted to I' by (-1.5, 4, 0.73) I' is decrypted to I'' by (-1.5, 4, **0.7300001**) I' and I'' are **99.620146**% different

the results that the proposed technique is competent in resisting histogram attacks.

5.6 Correlation coefficient analysis

The original image bears a high correlation among the neighboring pixels- taken in a horizontal, vertical, or diagonal direction. The goal is to minimize the adjacent pixels' correlation of an image encryption technique. A standard image encryption algorithm is marked as good if correlation nearly equal to zero is obtained for the encrypted images. The correlation coefficient of an image is calculated using (2). Many encryption techniques available in the literature have used position permutation at pixel and/or bit-level to reduce correlation. In the proposed method this is achieved by pixel substitution and iterative addition with the cyclic shift method.

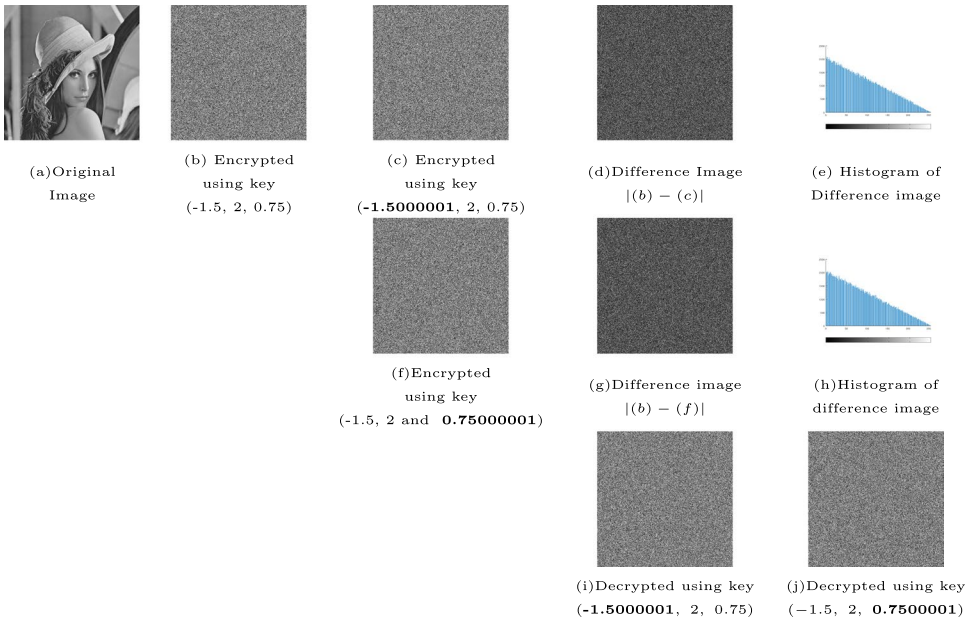


Fig. 6 Key sensitivity analysis

Image	Histogram				
	Original	Encrypted			
		P_{31}	P_{32}	P_{41}	P_{42}
Baboon					
Cameraman					
Lena					
Peppers					
Black					
Checkerboard					
Chest X-RAY					
Brain MRI					

Fig. 7 Histograms of Original and encrypted image

$$r_{X,Y} = \frac{\frac{1}{N}(\sum_{i=1}^{N}(x_i - E(X))(y_i - E(Y))}{\sqrt{D(X) * D(Y)}}$$
$$D(X) = \frac{\sum_{i=1}^{N}(x_i - E(X))^2}{N} \quad (2)$$
$$E(X) = \frac{\sum_{i=1}^{N}(x_i)}{N}$$

For the experiment, 1000 random locations are selected for each of the test images. Correlation coefficients of the original and its corresponding encrypted images are computed using (2) for the same randomly selected pair of pixels in each direction for each case. Due to randomly selected pixel pairs, correlation coefficients do not come the same in each run for any particular image. Here the average of absolute correlation coefficients received from ten trials for each of the test images and its corresponding encrypted images in horizontal, vertical, and diagonal direction are presented in Table 10. It is noted that the results listed in Table 10 are from encrypted images using polynomial P_{31} (due to space problem the results achieved using all the polynomials are not listed. The results for all the cases are nearly equal to the results presented). From the data listed in Table 10 it is clear that the original image holds a higher correlation coefficient, whereas its encrypted form has a correlation coefficient nearly equal to zero. The acceptance of the proposed method is signified from the obtained results.

Figure 8 shows the scatter diagrams of horizontally, vertically and diagonally adjacent pairs for 'Lena', 'Baboon', 'Cameraman', 'Peppers', 'Checkerboard', 'Black', 'Chest X-RAY' and 'Brain MRI' with their encrypted images achieved from polynomial P_{31}. Diagonally densely populated dots in the case of the original image signify higher correlation whereas scattered dots over the area for the encrypted images reflect very low or no correlation. Thus the scatter diagrams indicate a good performance of the scheme.

For comparison purposes, we have computed the average correlations with respect to the polynomials P_{31}, P_{32}, P_{41}, P_{42} and same set of selected pixels. This average result is reported in Table 11. Comparison of the proposed technique with the state-of-the-art proposals reflects the better performance of the proposed methods in most of the cases than the existing methods as displayed in Table 11.

Table 10 Correlation coefficient of original and encrypted images using P_{31}

Serial No.	Image	Image Type	Correlation Coefficients using P_{31}		
			Horizontal	Vertical	Diagonal
1	Baboon	Original	0.7454804	0.8571364	0.7629045
		Encrypted	0.0004133	0.0017512	0.0004269
2	Cameraman	Original	0.9864528	0.9802559	0.9697093
		Encrypted	0.0007204	0.0012293	0.0004658
3	Checkerboard	Original	0.9740539	0.9739948	0.9439421
		Encrypted	0.0008763	0.001557	0.0009915
4	Houses	Original	0.9211058	0.9131392	0.8492837
		Encrypted	0.0008859	0.0015468	0.0008472
5	Jetplane	Original	0.9617459	0.9698518	0.9505144
		Encrypted	0.0006977	0.0008948	0.0017175
6	Lake	Original	0.9686572	0.9752259	0.9662922
		Encrypted	0.0017097	0.0006964	0.0014504
7	Lena	Original	0.9865150	0.9757514	0.9647139
		Encrypted	0.0017702	0.0013383	0.0005553
8	Pepper	Original	0.9839224	0.9786951	0.9717631
		Encrypted	0.0012573	0.0007741	0.0019880
9	Barbara	Original	0.9428547	0.8877464	0.0003188
		Encrypted	0.0008285	0.0004818	0.0009148
10	Black	Original	0.9168121	0.8457116	0.8689953
		Encrypted	0.0009529	0.0008143	0.0007654
11	Chest X-RAY	Original	0.9953664	0.9928679	0.9913239
		Encrypted	0.0152276	0.0035175	0.0063179
12	Brain MRI	Original	0.9168105	0.8457102	0.8689908
		Encrypted	0.0004017	0.0136003	0.0009547

Table 11 Correlation coefficient analysis

Test Image	Direction	State-of-the-art Methods				Proposed
		Diaconu et al. [20]	Zhang et al. [31]	Kumar et al. [60]	Kandar et al. [56]	
Baboon	Horizontal	0.000833	0.022540	0.029000	0.001235	0.0004337
	Vertical	0.002817	0.009395	0.010090	0.001037	0.0014751
	Diagonal	0.001595	0.025113	0.009900	0.002015	0.0006482
Lena	Horizontal	0.003620	0.000944	0.008400	0.000946	0.001853
	Vertical	0.006488	0.024064	0.001800	0.000844	0.0019843
	Diagonal	0.003558	0.041171	0.000227	0.002741	0.0007435
Peppers	Horizontal	0.001859	0.046131	0.003400	0.001364	0.0011804
	Vertical	0.011580	0.026099	0.013000	0.000504	0.00072117
	Diagonal	0.001330	0.046131	0.005000	0.002817	0.0017840

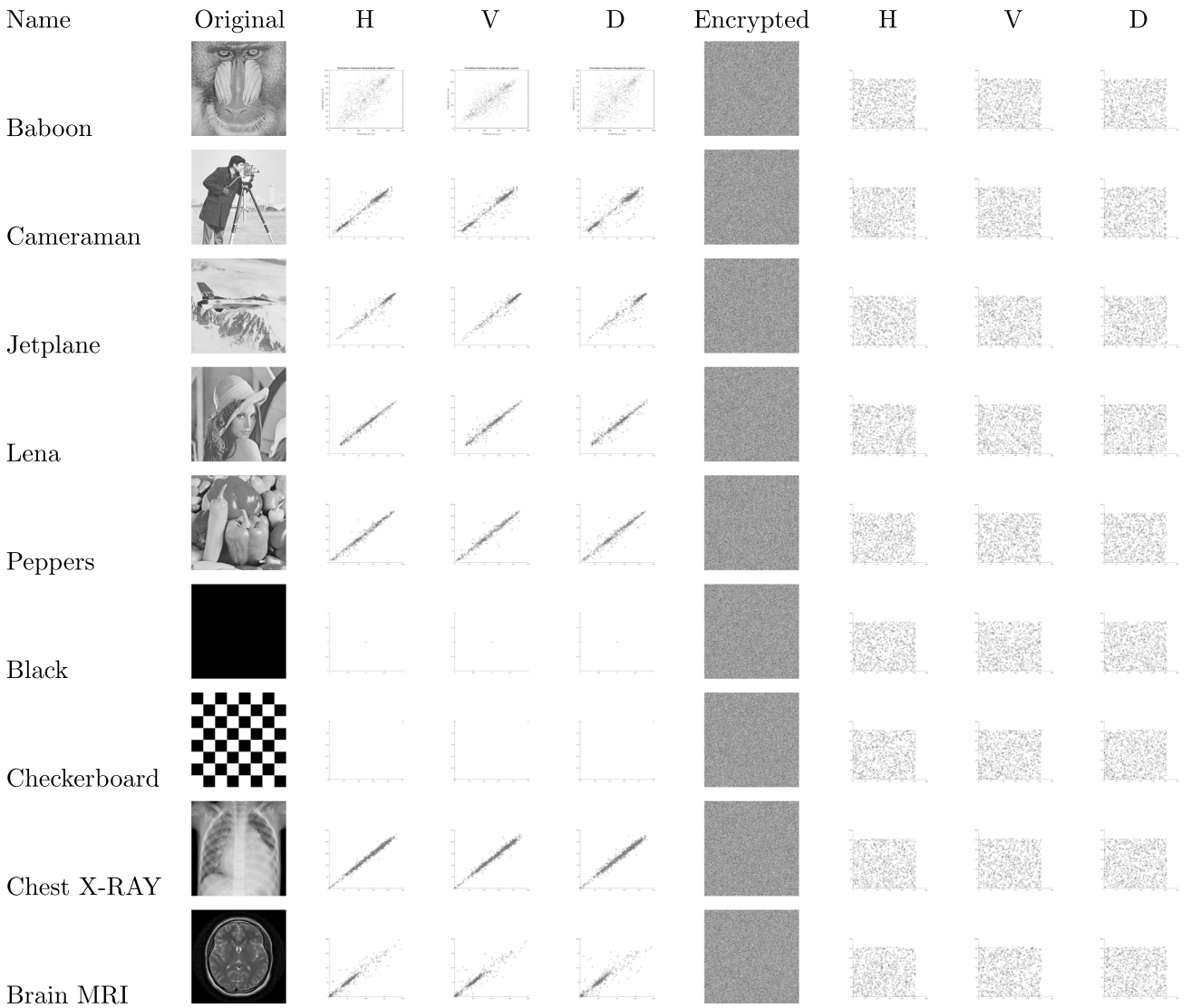


Fig. 8 Scatter diagram of original and encrypted images using P_{31}

5.7 Differential attack

To find the nature of the encryption algorithm, an attacker makes a small change of the original image; even a single bit and feeds both the original and the modified images in the encryption algorithm separately. This technique is called differential attack. **N**umber of **P**ixel **C**hange **R**ate (NPCR) and **U**nified **A**verage **C**hanging **I**ntensity (UACI) as denoted in (3) are the widely used techniques to measure the influence of differential attack. Here I_1 is the encrypted image obtained from the original and I_2 is from the modified image. The ideal value of NPCR is 100% and that of UACI is 33.33%.

$$NPCR(I_1,I_2) = \frac{1}{w*h} * \sum_{i=1}^{h}\sum_{j=1}^{w} D(i,j) * 100$$

$$UACI(I_1,I_2) = \frac{1}{w*h*255} * \sum_{i=1}^{h}\sum_{j=1}^{w} |I_1(i,j) - I_2(i,j)| * 100$$

$$D(i,j) = \begin{cases} 1, & \text{if } I_1(i,j) = I_2(i,j) \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

Here the LSB of the last pixel of the original image is changed to generate a modified image and both of them are passed through the encryption algorithm for each of the test images. NPCR and UACI are calculated over the pair of encrypted images. From the results listed in Table 12 it is clear that the NPCR and UACI values are very close to the ideal values. This signifies the immunity of the proposed schemes against differential attacks.

To have more test analysis 'Lena' image is modified to five different images only by changing LSB of first [(0,0)], middle [(256, 256)], last [(512,512)] and two random locations [(50, 450), (490, 10)] pixels values. The results of NPCR and UACI received using polynomial P_{31} are presented in Table 13. The comparative analysis of average NPCR and UACI of the proposed technique with the state-of-the-art methods reflects better performance in most of the cases as presented in Table 14.

5.8 Different attack model for cryptanalysis

For any encryption technique, it is assumed that the intruder has complete knowledge about the encryption algorithm but does not have information about the key used. S/He tries to compromise the encryption system by different types of attack models such as

i) Ciphertext only attack
ii) Known plain text attack
iii) Chosen plaintext attack
iv) Chosen ciphertext attack

In this subsection, the immunity of the proposed encryption system is discussed against the kind of attacks.

i) Ciphertext only attack: In this attack, the intruder has access only to the ciphertext (may access from the transmission channel) and from that, it tries to retrieve the plaintext or key information. The noisy cipher image

Table 13 NPCR and UACI values of encrypted Lena image using P_{31} for five modified pixel position

	Modified	Pixel Value		NPCR	UACI
	Pixel Position	Existing	Modified		
1	(0, 0)	162	163	99.9977	33.3389
1	(50, 450)	134	135	99.9985	33.4287
3	(256, 256)	104	105	99.8528	33.3123
4	(490, 10)	103	102	99.9496	33.3436
5	(512, 512)	108	109	99.6441	33.3887

Table 12 NPCR and UACI of the test images by modifying LSB of last pixel

Image	P_{31}		P_{32}		P_{41}		P_{42}	
	NPCR	UACI	NPCR	UACI	NPCR	UACI	NPCR	UACI
Baboon	99.6769	33.3679	99.9378	33.3979	99.6494	33.3330	99.9710	33.3426
Barbara	99.6662	33.3474	99.9454	33.4314	99.6452	33.3479	99.9638	33.4816
Cameraman	99.6742	33.2766	99.9352	33.5962	99.6479	33.4022	99.9638	33.3942
Houses	99.6643	33.3073	99.9325	33.5137	99.6464	33.2513	99.9687	33.2992
Jetplane	99.6693	33.2832	99.9367	33.4852	99.6624	33.3684	99.9657	33.4572
Lake	99.6605	33.3558	99.9435	33.4826	99.6521	33.3311	99.9695	33.3920
Lena	99.6441	33.3887	99.9443	33.4692	99.6361	33.4098	99.9672	33.4677
Pepper	99.5899	33.2898	99.9424	33.4728	99.6483	33.3677	99.9687	33.6110
Black	99.6723	33.3789	99.9426	33.4224	99.6449	33.3545	99.9607	33.4677
Checkerboard	99.6651	33.3208	99.9424	33.5670	99.6201	33.3207	99.9657	33.5273
Chest X-RAY	99.6624	33.3818	99.9423	33.6058	99.6201	33.2820	99.9695	33.4257
Brain MRI	99.6548	33.3106	99.9416	33.4925	99.6571	33.3692	99.9630	33.4090

Table 14 Comparison of plaintext sensitivity of proposed method with state-of-the-art techniques

		State-of-the-art Techniques				Proposed (Average)
		Diaconu et al. [20]	Zhang et al. [31]	Kumar et al. [60]	Kandar et al. [56]	
Lena	NPCR	99.5705	99.6088	99.6419	99.6208	99.7979
	UACI	33.4781	33.4671	33.5582	33.3183	33.4338
Baboon	NPCR	99.5943	99.6100	99.6153	99.6850	99.8088
	UACI	33.4905	33.4643	33.5785	33.4528	33.3604
Peppers	NPCR	99.5884	99.6089	99.6196	99.6839	99.7873
	UACI	33.5134	33.4689	33.5377	33.4504	33.4353

along with its entropy, correlation, and histogram indicates that the cipher image is a kind of random image. So, nothing can be guessed about the plaintext image from the cipher image. The attacker will fail to get key information as the size of the keyspace is significantly large.

ii) Known plaintext attack: In this kind of attack, the attacker has gained (maybe from some previous communication) some plenty amount of plaintexts and its corresponding ciphertexts. From those, it tries to get information of the key used or tries to retrieve the plaintext from the currently captured ciphertext. The proposed technique is highly sensitive to the key (see Fig. 6) and has immunity against differential attack (see Table 12). Thus from the acquired information of plaintext, ciphertext of earlier communications it is hard to retrieve the plaintext of current communication.

iii) Chosen plaintext attack: Here the attacker has temporary access to the encryption system and it tries with some combinations of plaintext and ciphertext to find a match with the received ciphertext. The proposed technique has a large keyspace (see Section 5.1), thus the brute-force attack will fail. The attacker may try to use some temporary vectors like the permutation used in Algorithm 3 and/or Algorithm 4. The permutation used in Algorithms 3 and 4 have 256 and h (height of the image) entry thus 256! and h! different combination of permutations are there respectively. To successfully retrieve the plaintext, the attacker has to opt $256! \times h!$ different trails. It is infeasible for an attacker to opt for the trails in some bounded time frame even by using a supercomputer having the capacity of calculation 2^{80} instructions per second.

iv) Chosen ciphertext attack: It is the same as (iii), but here the attacker has temporary access to the decryption algorithm and it tries with some possible combination of ciphertext (or some possible combination of keys on ciphertext) to retrieve the plaintext. This technique will fail due to the large keyspace and a huge number of trials of the permutations.

From the above discussion, it can be concluded that the proposed technique has resistance against different attacks of cryptoanalysis.

6 Conclusions

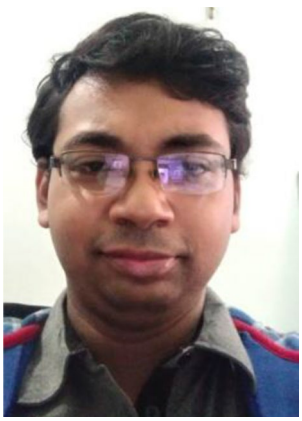
An image encryption proposal using a non-chaotic technique is presented in this paper. Modified Regula-Falsi method is used to generate a pseudo-random sequence which is further applied to 'define permutation in the image encryption technique. Pixel value substitution and addition with the cyclic shift are used to encrypt an image. The technique requires only $(256 + h)$ length sequence where h is the height of the image. Good experimental results in terms of noise like encrypted images provide evidence of good performance. From security analysis, it is found that the proposed method is secure against different kinds of attacks. Comparison of the proposed technique with state-of-the-art methods signifies its acceptability as an image encryption technique.

References

1. Chen J-x, Zhu Z-l, Fu C, Zhang L-b, Zhang Y (2015) An image encryption scheme using nonlinear inter-pixel computing and swapping based permutation approach. Communications in Nonlinear Science and Numerical Simulation 23(1–3):294–310
2. Arab A, Rostami MJ, Ghavami B (2019) An image encryption method based on chaos system and aes algorithm. The Journal of Supercomputing 75(10):6663–6682
3. Pareek NK, Patidar V, Sud KK (2006) Image encryption using chaotic logistic map. Image and Vision Computing 24(9):926–934
4. Bourbakis N, Alexopoulos C (1992) Picture data encryption using scan patterns. Pattern Recognition 25(6):567–581
5. Kuo CJ (1993) Novel image encryption technique and its application in progressive transmission. Journal of Electronic Imaging 2(4):345–352
6. Li X, Knipe J, Cheng H (1997) Image compression and encryption using tree structures. Pattern Recognition Letters 18(11–13):1253–1259

7. Chang HK-C, Liu J-L (1997) A linear quadtree compression scheme for image encryption. Signal Processing: Image Communication 10(4):279–290
8. Refregier P, Javidi B (1995) Optical image encryption based on input plane and fourier plane random encoding. Optics Letters 20(7):767–769
9. Matthews R (1989) On the derivation of a chaotic encryption algorithm. Cryptologia 13(1):29–42
10. Fridrich J (1998) Symmetric ciphers based on two-dimensional chaotic maps. International Journal of Bifurcation and chaos 8(06):1259–1284
11. Sathishkumar G, Srinivas R, Bagan KB (2012) Image encryption using random pixel permutation by chaotic mapping. In: 2012 IEEE symposium on computers & informatics (ISCI). IEEE, pp 247–251
12. Zhang X, Zhao Z (2014) Chaos-based image encryption with total shuffling and bidirectional diffusion. Nonlinear Dynamics 75(1–2):319–330
13. Gao T, Chen Z (2008) A new image encryption algorithm based on hyper-chaos. Physics Letters A 372(4):394–400
14. Li H, Wang Y, Yan H, Li L, Li Q, Zhao X (2013) Double-image encryption by using chaos-based local pixel scrambling technique and gyrator transform. Optics and Lasers in Engineering 51(12):1327–1331
15. Li Y, Wang C, Chen H (2017) A hyper-chaos-based image encryption algorithm using pixel-level permutation and bit-level permutation. Optics and Lasers in Engineering 90:238–246
16. Solak E, Çokal C, Yildiz OT, Biyikoğlu T (2010) Cryptanalysis of fridrich's chaotic image encryption. International Journal of Bifurcation and Chaos 20(05):1405–1413
17. Zhu C, Liao C, Deng X (2013) Breaking and improving an image encryption scheme based on total shuffling scheme. Nonlinear Dynamics 71(1–2):25–34
18. Teng L, Wang X, Meng J (2018) A chaotic color image encryption using integrated bit-level permutation. Multimedia Tools and Applications 77(6):6883–6896
19. Zhang W, Yu H, Zhao Y-l, Zhu Z-l (2016) Image encryption based on three-dimensional bit matrix permutation. Signal Processing 118:36–50
20. Diaconu A-V (2016) Circular inter-intra pixels bit-level permutation and chaos-based image encryption. Information Sciences 355:314–327
21. Sun S (2018) A novel hyperchaotic image encryption scheme based on dna encoding, pixel-level scrambling and bit-level scrambling. IEEE Photonics Journal 10(2):1–14
22. Ping P, Fan J, Mao Y, Xu F, Gao J (2018) A chaos based image encryption scheme using digit-level permutation and block diffusion. IEEE Access 6:67581–67593
23. Shahna K, Mohamed A (2020) A novel image encryption scheme using both pixel level and bit level permutation with chaotic map. Appl Soft Comput 90:106162
24. Idrees B, Zafar S, Rashid T, Gao W (2019) Image encryption algorithm using s-box and dynamic hénon bit level permutation. Multimed Tools Appl: 1–28
25. Biswas P, Kandar S, Dhara BC (2017) A novel image encryption technique using one dimensional chaotic map and circular shift technique. In: Proceedings of the 6th international conference on software and computer applications. pp 112–116
26. Zhu Z-l, Zhang W, Wong K-w, Yu H (2011) A chaos-based symmetric image encryption scheme using a bit-level permutation. Information Sciences 181(6):1171–1186
27. Zhou Y, Cao W, Chen CP (2014) Image encryption using binary bitplane. Signal Processing 100:197–207
28. Wang P, Qiu J (2019) An adaptive image encryption scheme based on bit-level permutation. In: Proceedings of the international conference on artificial intelligence. Information Processing and Cloud Computing, pp 1–5
29. Borujeni SE, Eshghi M (2013) Chaotic image encryption system using phase-magnitude transformation and pixel substitution. Telecommunication Systems 52(2):525–537
30. Hua Z, Zhou Y, Pun C-M, Chen CP (2015) 2d sine logistic modulation map for image encryption. Information Sciences 297:80–94
31. Zhang Y (2018) The unified image encryption algorithm based on chaos and cubic s-box. Information Sciences 450:361–377
32. Çavuşoğlu Ü, Kaçar S, Pehlivan I (2017) Zengin A Secure image encryption algorithm design using a novel chaos based s-box. Chaos, Solitons & Fractals 95:92–101
33. Hussain I, Gondal MA (2014) An extended image encryption using chaotic coupled map and s-box transformation. Nonlinear Dynamics 76(2):1355–1363
34. Liu Y, Tong X, Ma J (2016) Image encryption algorithm based on hyper-chaotic system and dynamic s-box. Multimedia Tools and Applications 75(13):7739–7759
35. Farah MB, Farah A, Farah T (2019) An image encryption scheme based on a new hybrid chaotic map and optimized substitution box. Nonlinear Dyn: 1–24
36. Lu Q, Zhu C, Deng X (2020) An efficient image encryption scheme based on the lss chaotic map and single s-box. IEEE Access
37. Mousavi M, Sadeghiyan B (2021) A new image encryption scheme with feistel like structure using chaotic s-box and rubik cube based p-box. Multimedia Tools and Applications 80(9):13157–13177
38. Laiphrakpam DS, Khumanthem MS (2017) Medical image encryption based on improved elgamal encryption technique. Optik 147:88–102
39. Hua Z, Yi S, Zhou Y (2018) Medical image encryption using high-speed scrambling and pixel adaptive diffusion. Signal Processing 144:134–144
40. Cao W, Zhou Y, Chen CP, Xia L (2017) Medical image encryption using edge maps. Signal Processing 132:96–109
41. Nematzadeh H, Enayatifar R, Motameni H, Guimarães FG, Coelho VN (2018) Medical image encryption using a hybrid model of modified genetic algorithm and coupled map lattices. Optics and Lasers in Engineering 110:24–32
42. Chai X, Zhang J, Gan Z, Zhang Y (2019) Medical image encryption algorithm based on latin square and memristive chaotic system. Multimedia Tools and Applications 78(24):35419–35453
43. Thakur S, Singh A, Kumar B, Ghrera S (2020) Improved dwt-svd-based medical image watermarking through hamming code and chaotic encryption. In: Advances in VLSI, communication, and signal processing. Springer, pp 897–905
44. Jeevitha S, Prabha NA (2021) Novel medical image encryption using dwt block-based scrambling and edge maps. Journal of Ambient Intelligence and Humanized Computing 12(3):3373–3388
45. Ravichandran D, Murthy B, Balasubramanian V, Fathima S, Amirtharajan R et al (2021) An efficient medical image encryption using hybrid dna computing and chaos in transform domain. Medical & Biological Engineering & Computing 59(3):589–605
46. Babaei A, Motameni H, Enayatifar R (2020) A new permutation-diffusion-based image encryption technique using cellular automata and dna sequence. Optik 203:164000
47. Enayatifar R, Guimarães FG, Siarry P (2019) Index-based permutation-diffusion in multiple-image encryption using dna sequence. Optics and Lasers in Engineering 115:131–140
48. Azimi Z, Ahadpour S (2019) Color image encryption based on dna encoding and pair coupled chaotic maps. Multimed Tools Appl: 1–18

49. Guesmi R, Farah MAB, Kachouri A, Samet M (2016) A novel chaos-based image encryption using dna sequence operation and secure hash algorithm sha-2. Nonlinear Dynamics 83(3):1123–1136
50. Kaur M, Singh D, Sun K, Rawat U (2020) Color image encryption using non-dominated sorting genetic algorithm with local chaotic search based 5d chaotic map. Futur Gener Comput Syste
51. Chen J-x, Zhu Z-l, Fu C, Yu H, Zhang L-b (2015) An efficient image encryption scheme using gray code based permutation approach. Optics and Lasers in Engineering 67:191–204
52. Sinha RK, Agrawal I, Jain K, Gupta A, Sahu S (2020) Image encryption using modified rubik's cube algorithm. In: Advances in computational intelligence. Springer, pp 69–78
53. Vidhya R, Brindha M (2020) A chaos based image encryption algorithm using rubik's cube and prime factorization process (cierpf). J King Saud Univ-Comput Inf Sci
54. Das SK, Dhara BC (2017) A new image encryption method using circle. In: 2017 8th International conference on computing, communication and networking technologies (ICCCNT). IEEE, pp 1–6
55. Priya A, Sinha K, Darshani MP, Sahana SK (2019) A novel multimedia encryption and decryption technique using binary tree traversal. In: Proceeding of the second international conference on microelectronics, computing & communication systems (MCCS 2017). Springer, pp 163–178
56. Kandar S, Chaudhuri D, Bhattacharjee A, Dhara BC (2019) Image encryption using sequence generated by cyclic group. Journal of Information Security and Applications 44:117–129
57. Biswas P, Kandar S, Dhara BC (2020) An image encryption scheme using sequence generated by interval bisection of polynomial function. Multimedia Tools and Applications 79(43):31715–31738
58. Rukhin A, Soto J, Nechvatal J, Smid M, Barker E (2001) A statistical test suite for random and pseudorandom number generators for cryptographic applications. Tech. rep, Booz-allen and hamilton inc mclean va
59. Brown RG, Eddelbuettel D, Bauer D (2018) Dieharder Duke University Physics Department Durham, NC 27708–0305
60. Kumar CM, Vidhya R, Brindha M (2021) An efficient chaos based image encryption algorithm using enhanced thorp shuffle and chaotic convolution function. Appl Intell: 1–30



Aakash Paul Did his master from Indian Institute of Engineering Science and Technology, Shibpur (IIEST, Shibpur). His research interests are image and video encryption, secret sharing, digital watermarking etc.

Shyamalendu Kandar Currently working an Assistant Professor in the Department of Information Technology, Indian Institute of Engineering Science and Technology (an Institute of National Importance), Shibpur, India. He has obtained his M.Tech. in Information Technology from Jadavpur University and Ph.D from the same university. He has contributed a number of research papers in several peer reviewed international journals and conferences. He is the author of two books on Automata theory. His research interests are Secret sharing, Visual Cryptography, Image encryption, Remote user authentication, Machine learning etc.

Bibhas Chandra Dhara Professor of Department of Information Technology of Jadavpur University. Did his B.Tech from Calcutta University and Master and Ph.D from ISI, Kolkata. His is the author of several peer reviewed international journals and conferences of repute. His research interest are Algorithms, Security, Image and Video processing, Pattern Recognition etc.