3D Res (2017) 8:37
DOI 10.1007/s13319-017-0148-5

3DR REVIEW

A Survey of Image Encryption Algorithms

Manju Kumari · Shailender Gupta · Pranshul Sardana

Received: 19 August 2017 / Revised: 10 October 2017 / Accepted: 30 October 2017 / Published online: 13 November 2017

Abstract Security of data/images is one of the crucial aspects in the gigantic and still expanding domain of digital transfer. Encryption of images is one of the well known mechanisms to preserve confidentiality of images over a reliable unrestricted public media. This medium is vulnerable to attacks and hence efficient encryption algorithms are necessity for secure data transfer. Various techniques have been proposed in literature till date, each have an edge over the other, to catch-up to the ever growing need of security. This paper is an effort to compare the most popular techniques available on the basis of various performance metrics like differential, statistical and quantitative attacks analysis. To measure the efficacy, all the modern and grown-up techniques are implemented in MATLAB-2015. The results show that the chaotic schemes used in the study provide highly scrambled encrypted images having uniform histogram distribution. In addition, the encrypted images provided very less degree of correlation coefficient values in horizontal, vertical and diagonal directions, proving their resistance against statistical attacks. In addition, these schemes are able to resist differential attacks as these showed a high sensitivity for the initial conditions, i.e. pixel and key values. Finally, the schemes provide a large key spacing, hence can resist the brute force attacks, and provided a very less computational time for image encryption/decryption in comparison to other schemes available in literature.

Keywords Chaotic function · Cryptography · Differential attacks · Encryption/decryption · Statistical attacks · S-box

1 Introduction

The technological advancements of last two decade have provided the world with systems which can transmit large chunks of data, like images, efficiently via the generic public networks. These public routes, though are reliable pathways, are vulnerable to be accessed by unauthenticated users and the data transferring through them can be accessed and altered by unauthenticated user access like ransom ware. Figure 1 shows yearly data for annual number of approximate ransom ware families since 2004 with the projected increase in 2017 [1]. It is quite evident from the figure that the data transferred over public domain is quite susceptible to attacks. Hence, data should be transformed into a secure format using encryption while transmission. Image encryption enhances the

M. Kumari · S. Gupta (✉) · P. Sardana
YMCA University of Science and Technology, Faridabad, India
e-mail: manju_mrce027@yahoo.com

S. Gupta
e-mail: shailender81@gmail.com

P. Sardana
e-mail: pranshulsardana@rediffmail.com

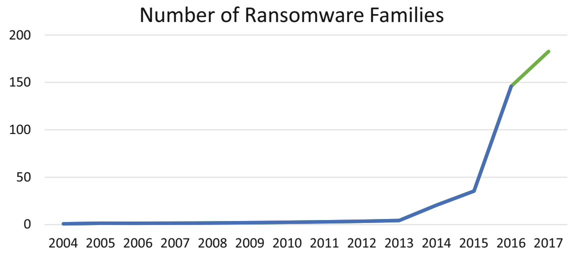


Fig. 1 Annual number of ransom ware families per year

security of digital images and is an essential component in various applications, like video conferencing, telemetric medical imaging systems and military image transmissions.

Various image encryption techniques are proposed worldwide [2–6] but many of the techniques are susceptible to attacks including differential and statistical attacks [7, 8]. The conventional encryption mechanisms like Advanced Encryption Standard, Blowfish, and International Data Encryption Algorithm are suitable for encryption of texts but, these technique turn out to be ineffective when used for image encryption application [9–12]. This is due to the intrinsic characteristic of the images like strong correlation, high redundancy and high computational expense.

Therefore researcher focused [6–15] on devising a mechanism for image encryption that has the following characteristics:

- *Low Correlation* The correlation between the original and the encrypted image should be as low as possible. Ideally it should be zero.
- *Large Key Space* The key size should be large since more is the key space, higher is the brute force search time.
- *Key Sensitivity* The algorithm should have high key sensitivity or in other words a slight change in the key value should change significantly the encrypted image.
- *Entropy* It is a measure of randomness. In ideal case the entropy should be eight.
- *Low Time Complexity* An algorithm that has high computational time is not feasible for practical applications. Therefore, an image encryption algorithm should have low time complexity.

This paper is an effort to compare numerous classical to modern cryptography algorithms based on various performance parameters like time complexity, key sensitivity and entropy values. Moreover, the performance of all these algorithms under statistical, differential and qualitative attacks is also analyzed.

The rest of the paper is organized as follows: Section two reviews all the fifteen techniques used in the study. Block diagram representation of all the techniques is also provided for an easier understanding of the flows of the processes. Third section summarizes the simulation setup parameters which contains the setup parameters and performance metrics used. The performance metrics is based on visual assessment, statistical analysis, differential attack analysis, key space analysis, key sensitivity analysis, quantitative analysis and time complexity. The fourth section compares the results obtained by applying the encryption schemes used in the study based on the performance metrics. Finally, the fifth section concludes the study by comparing and summarizing all the results obtained.

2 Literature Review

In the field of cryptography, various classic and modern algorithms have been explored, each with their own advantages and disadvantages. Explanation of the fifteen schemes computed in the study is given below.

2.1 Vigenère Cipher

Vigenère cipher is a kind of poly alphabetic cipher that consists of a series of different Cesar ciphers for encryption [16]. It uses a series of elements in tabular form, also known as the Vigenère table. In this table, the first row consists of n different elements and the remaining table has n-1 similar rows each proceeding one formed by left cycle shifting of elements of previous row. Decryption can be done by looking up the ciphered element in the row corresponding to the key element, and then the column will represent the decrypted output, i.e. the original letter. Conventionally, the Vigenère cipher was developed to encrypt alphabetical texts by using the Vigenère table, but many of the recent studies have used the concepts of Vigenère cipher for image encryption. These techniques used both chaotic and non-chaotic based encryption schemes. Figure 2 shows the block diagram of Vigenère cipher. It takes an input image and encrypts it using Vigenère table and key. The key value is repeated until all the image pixels are encrypted.

2.2 Data Encryption Standard (DES)

Data Encryption Standard (DES) is one of earliest block ciphers developed in 1970s, at IBM and later adopted by National Bureau of Standards [17, 18]. It takes an input block of 64 bit (8 pixels at a time) and applies Initial Permutation (IP) to it. The permutated data is then divided into two sub blocks (Li, Ri). These sub-blocks after passing through sixteen round operations using different keys (48 bit) are finally permuted to obtain final cipher text. The function block shown in DES is a combination of Expansion permutation (32–48 bit), Xoring Operation followed by substitution (48–32 bit) and final straight permutation box. The initial key length is 64 bits, out of which 8 bits are reserved for parity checks. Out of the remaining 64 bits, 56 bits are extracted using PC1. This 56 bit data is now divided into two halves and rotated various times to obtain 16 sub keys (56 bit keys). From these 56 bit sub keys, 48 bits keys (16 keys for round operations) are extracted using PC2. Decryption follows a similar mechanism of rounds as encryption does, but with the order of sub keys is inverted. Even though the encryption and decryption processes proceeds in a high number of rounds, the DES security mechanism is breakable by many ways. Brute force attack and known-plain text attacks are the most common approaches [19]. Figure 3 shows the block diagram of DES. It shows how different sub keys are generated along with the encryption/decryption mechanism.

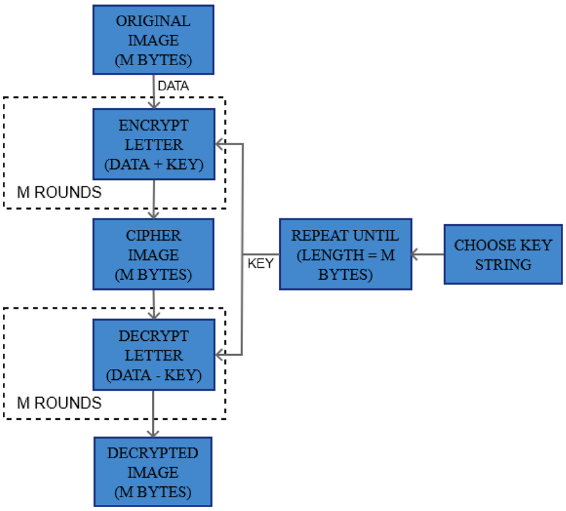

ORIGINAL IMAGE (M BYTES)
DATA
ENCRYPT LETTER (DATA + KEY)
M ROUNDS
CIPHER IMAGE (M BYTES)
KEY
REPEAT UNTIL (LENGTH = M BYTES)
CHOOSE KEY STRING
DECRYPT LETTER (DATA - KEY)
M ROUNDS
DECRYPTED IMAGE (M BYTES)

Fig. 2 Block diagram of Vigenère cipher

2.3 International Data Encryption Algorithm (IDEA)

International Data Encryption Algorithm (IDEA), also known as Improved Proposed Encryption Standard (IPES), is a symmetric-key block cipher and a successor of Proposed Encryption Standard (PES) [20]. It uses a fixed key size of 128-bits and an input block size of 64-bits. The block is then sub divided into four sub blocks (A, B, C, D). The encryption and decryption structures are similar and use eight full rounds plus an additional half-round, making a total of 8.5 rounds. Various components included in each round are Bitwise Exclusive-OR (EX-OR), Adder and Multipliers (see Fig. 4). Each round uses a total of six keys, while the half-round uses four keys for both encryption and decryption processes. So, IDEA uses a total of 52 sub keys which are obtained directly from the initial key. The 128 bit key is divided into 8 sub keys. After every round 25-bit left rotation of the key is performed and new sub keys are obtained as shown in Fig. 4. The encryption sub keys (S[i]) in the quadratic clusters, like S[0] to S[3], are substituted in an inverted order, here S[51] to S[48]. While the paired sub keys, like S[4] and S[5], are directly substituted, here S[46] to S[47]. IDEA is vulnerable to various kind of attack like narrow-bicliques attack and man-in-the-middle attack.

2.4 Blowfish

Blowfish is a symmetric-key block cipher algorithm. Its main component is a Feistel network, iterating 16 times [21]. The size of the block used is same as DES algorithm (64 bits). But, unlike DES, it uses a variable key length of 32–448 bits. The block diagram shows that the 64 bit data (8 pixels) are divided into sub halves. These sub blocks pass through 16 rounds of operation using S box, adder and bit wise Exor functions of the Feistel structure as shown in the Fig. 5. The output of each round is an input to the next round. Finally, the left and right sub blocks are Xored

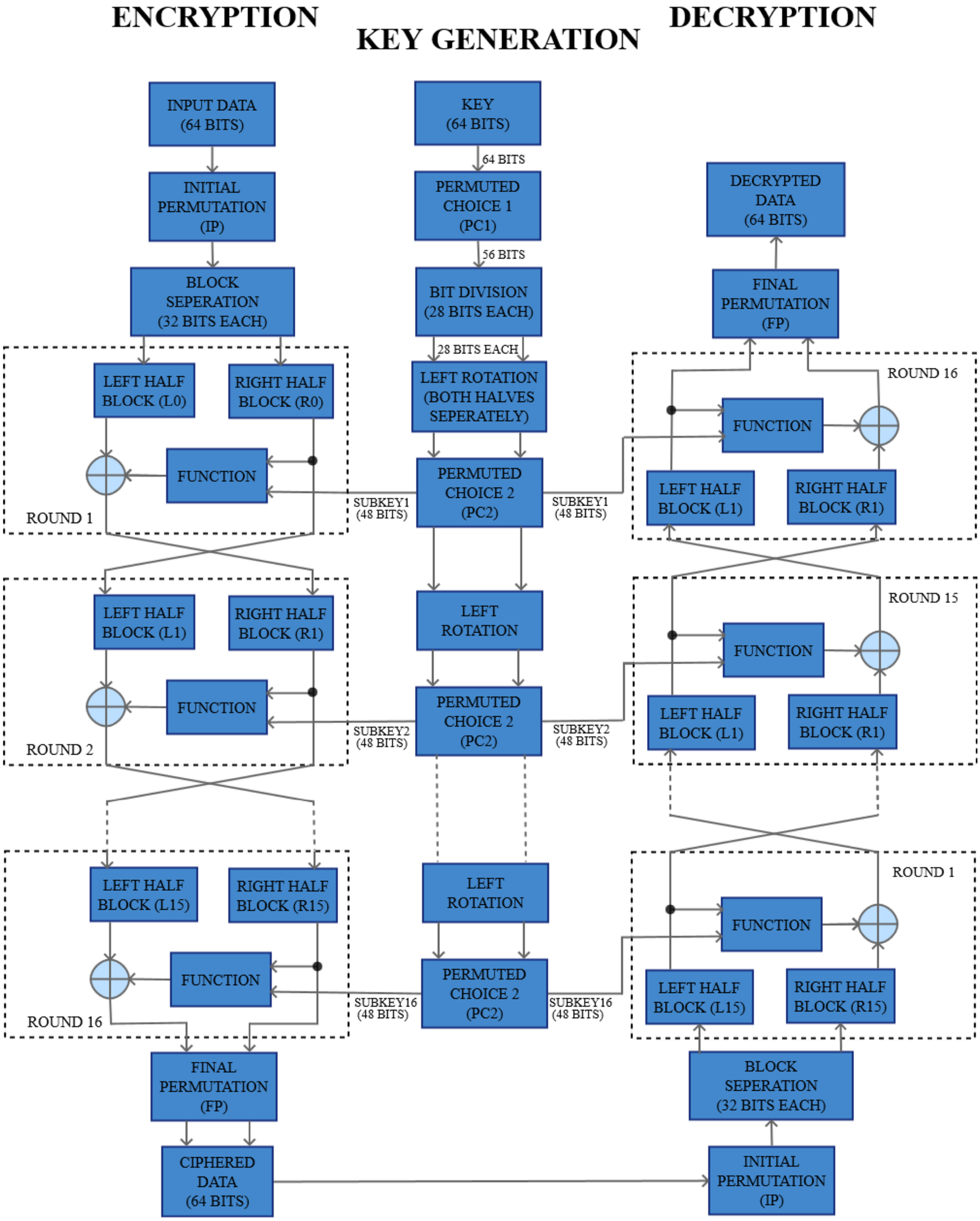


Fig. 3 Block diagram of Data Encryption Standard. ⊕ Bit wise Exor operation

with the key values (P ARRAY GENERATOR 17 and 18) and concatenated to obtain the final cipher text. Blowfish uses a relatively large key: a P array containing 18 key (32-bit) numbers and four S boxes, each with 256 entries initialized to random values. The next step is to XOR P-array with the key bits for

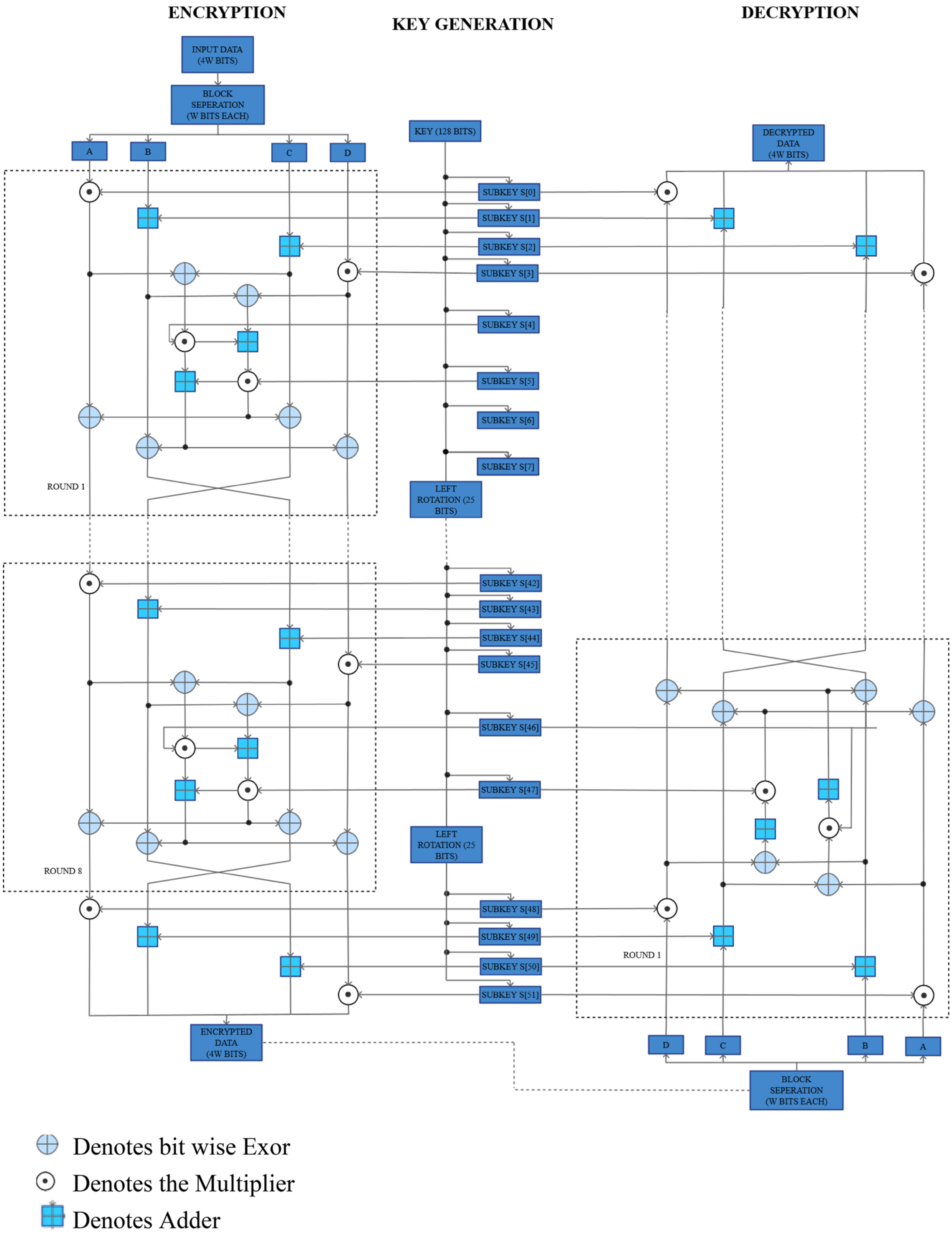


Fig. 4 Block diagram of International Data Encryption Algorithm. ⊕ Bit wise Exor. ⊙ The multiplier. ⊞ Adder

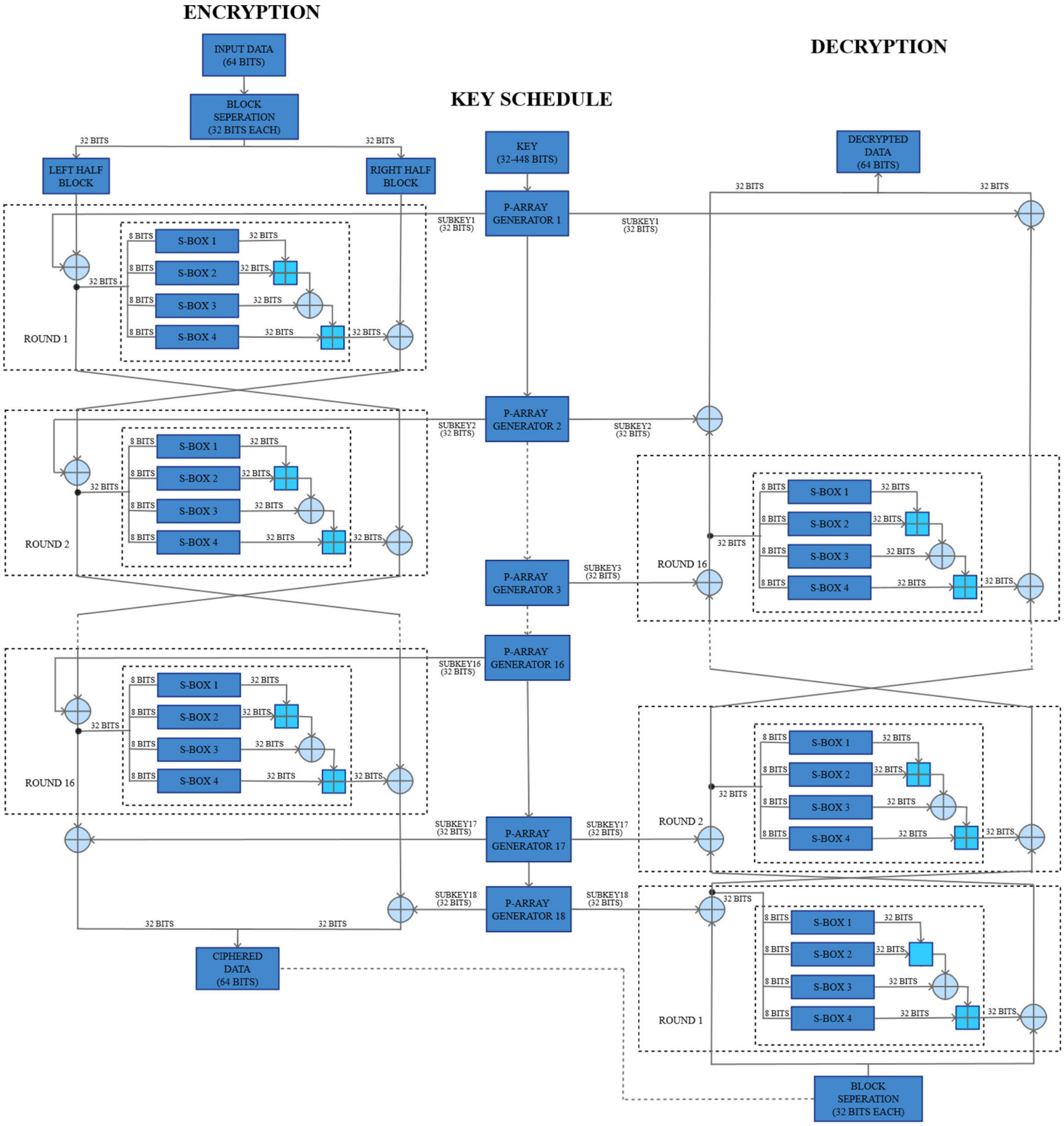


Fig. 5 Block diagram of Blowfish algorithm

example, P1 XOR (first 32 bits of key), P2 XOR (second 32 bits of key). In this way all zero string are encrypted. This resultant output is now P1 and P2. This P1 and P2 are not encrypted with the modified sub keys to obtain P3 and P4. The process is repeated to obtain all keys. Despite of having a convoluted initialization, there is an efficient encryption of data. Many of the blowfish algorithms are still unbroken as those are protected by patent laws but Blowfish has its own limitations too. Its utilization is constrained to applications, like communication links, in which key changes infrequently. Also, as Blowfish has small block size, file greater than 4 Gb are not recommended to be encrypted.

2.5 Visual Cryptography

Visual cryptography is the famous technique of hiding the secret messages, like images, objects or texts in a multiple share format [22]. The initial researches on visual cryptography were based on two share schemes and constituted only of Black (B) and White (W) components. In recent years, researchers have explored visual cryptography for grayscale and color images too. Along with that, the number of shares has also increased beyond two. It is a promising alternative above other image encryption schemes because instead of using complex algorithms for securing the data it uses the visual perception and hence adding an additional layer of security. For a two share Black/White (B/W) image encryption process, the two shares have pixel pairs in the order BW (Encrypted Share 1) or WB (Encrypted Share 2), which overlap to produce the resulting images. This overlap process is the simple Ex-XOR function as shown in Fig. 6. It is important to note her that until the hacker doesn't have both the share, the image decryption is not possible. Figure 6 shows the block diagram of visual cryptography.

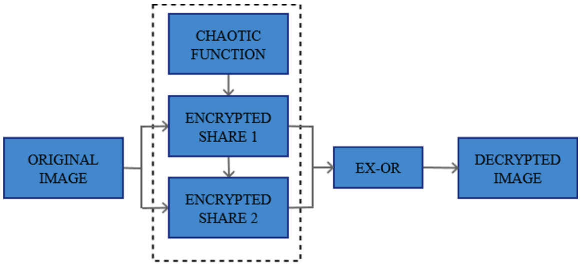


Fig. 6 Block diagram of visual cryptography

2.6 RC4

RC4 is a remarkably fast and simple symmetric key, stream cipher [18]. It was first designed in 1987 originally as a trade secret but was leaked a few years later in 1994. The algorithm uses a key length varying from 1 to 256 bytes (generally between 5 and 16 bytes). The key value is repeated to obtain a key of length 256 byte. Using this key, permutation function and randomized array, a pseudorandom byte is generated to encrypt the plaintext (bitwise XOR). The decryption process follows a similar mechanism (see Fig. 7). Regardless of its simplicity, speed and easy implementation, its applications are limited because it possesses several vulnerabilities. Such as, the Invariance Weakness, which uses numerous partial plaintexts recovery attacks to steal data like credit card numbers, passwords etc. In order to make RC4 robust against this and other type of attacks, various variants like RC4A, Spritz have been developed.

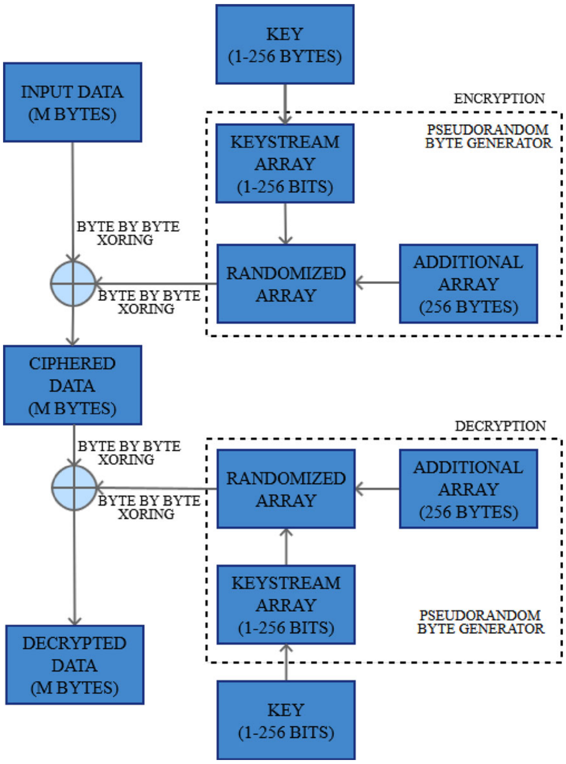


Fig. 7 Block diagram of RC4

2.7 RC5

RC5, a successor of RC4, is a symmetric-key block cipher [23]. Like RC4, this cipher uses simple logic. RC5 was also developed by Ron Rivest, seven years after the predecessor was developed, in 1994. The encryption and decryption processes are similar in implementation. The algorithm can use a key size varying from 0 to 2040 bits (with a recommended size of 128 bits), and the key schedule is more convoluted than the prior one. The RC5 encryption involves two major steps:

Key Generation: The key expansion algorithm uses two word sized binary constants Pw and Qw They are defined for arbitrary word w as follows:

$$Pw = odd((e - 2)2^{w})$$

$$Qw = odd((\varphi - 2)2^{w})$$

e is base of natural log while φ is golden ratio.

From these words a random Stream (S) is initialized of t words [24]. Using this random stream of words, adder, shifting and key value (converted to c word), a pseudorandom stream of Sub keys is generated. This key is used for encryption and decryption in round process.

Encryption/Decryption The cipher uses a variable block size which can be 32, 64, or 128 bits and also the number of rounds varying from 0 to 255 rounds. It is recommended to use 18–20 rounds for a fast and secure implementation. The block is divided into two halves (A and B). Finally, each block is passed through rounds using adder, left shift and ex-or operation. The output of one round is fed as an input to the next round. The reverse process is carried out at the receiver side to obtain the plain text data. A differential attack using 2^{44} chosen plaintext can break a 12-round, 64-bit block RC5 version (see Fig. 8).

2.8 RC6

RC6 [25], a symmetric-key block cipher, is a successor of the RC4 and RC5 algorithms. It was developed by Ron Rivest with others in 1998 and was a runner-up in the Advanced Encryption Standard (AES) challenge. The algorithm uses a key size of 128, 192, or 256-bits and uses a block size of 128-bits. It uses a Feistel network have the structure similar to RC5. Both the algorithms have a similar key expansion as they use a key of t words for subkeys, which is generated from the initial key. But, RC6 uses four w-bit word registers instead of two. It also includes elements like a quadratic equation and integer multiplication (F) as a part of the transformation. The extensive base build up by the previous algorithms and AES evaluations of RC5 helped towards designing a pretty good encryption algorithm and there is no practical attack which can breach RC6 is a sensible amount of time. Figure 9 shows the block diagram of RC6.

2.9 Triple Data Encryption Standard (TDES)

Triple Data Encryption Standard (TDES), also known as Triple Data Encryption Algorithm (TDEA), is a symmetric-key block cipher. As the name suggests, the algorithm utilizes the DES algorithm three times all in encryption, decryption and key generation processes. TDES uses three 56-bit keys for the encryption and decryption processes [26]. The process of selection of these keys is known as keying option. The TDES process provides three keying options as follows:

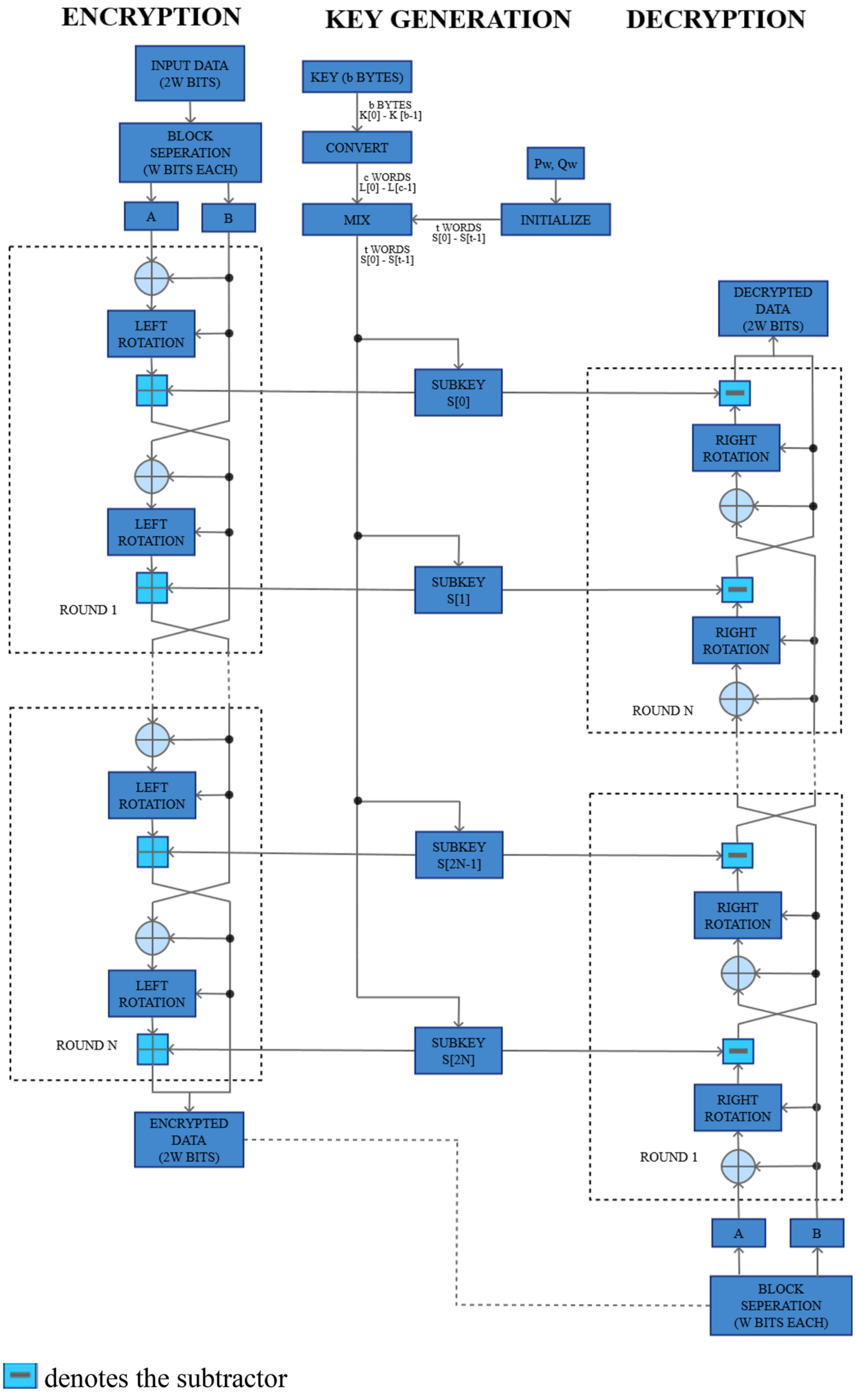
1. Keying option 1: All the three keys are independent from each other. It is the most reliable keying option and is not vulnerable to any known practical attacks.
2. Keying option 2: K1 and K2 are independent, while K3 is same as of K1. It is resistant against meet-in-the-middle attack but is vulnerable to attacks like chosen-plaintext. It is also known as 2DES
3. Keying option 3: All the three keys are identical. It the weakest keying option

Regardless of the keying option, the encryption processes of TDES uses K1 for encrypt K2 to decrypt and K3 to encrypt the data again. Similarly, the decryption processes uses the respective keys to decrypt, encrypt and decrypt the data. Figure 10 shows the block diagram of TDES.

2.10 Advanced Encryption Standard (AES)

AES is a symmetric-key algorithms belonging to the Rijndeal cipher family. Specifically, three different members of the family were adopted by National Institute of Standards and Technology, U.S. as AES. These had an equal block size of 128-bit but had varying key sizes of 128, 192, 256-bits signifying increase in security strength with increase in bits [27]. This increase in strength is a result of increase in number of cycles of repetition (rounds) as a higher bit AES is used. The respective number of cycles of repetition is 10, 12 and 14. AES utilizes a substitution-permutation network structure, while the previously widely used DES was based on Fiestel network. Different encryption and decryption processes uses similar byte substitution, shift row, mix column (except in round 10) and add round key steps. All three accepted versions of AES have a very similar key schedule and use a large number of sub keys, like for the 128-bit version 11 sub keys are used. Most versions of AES work utilizing a 4 × 4 matrix, which gives the cipher its non-linear effect and hence contributes towards the strength. A full brute force attack is the fastest documented attack and hence AES

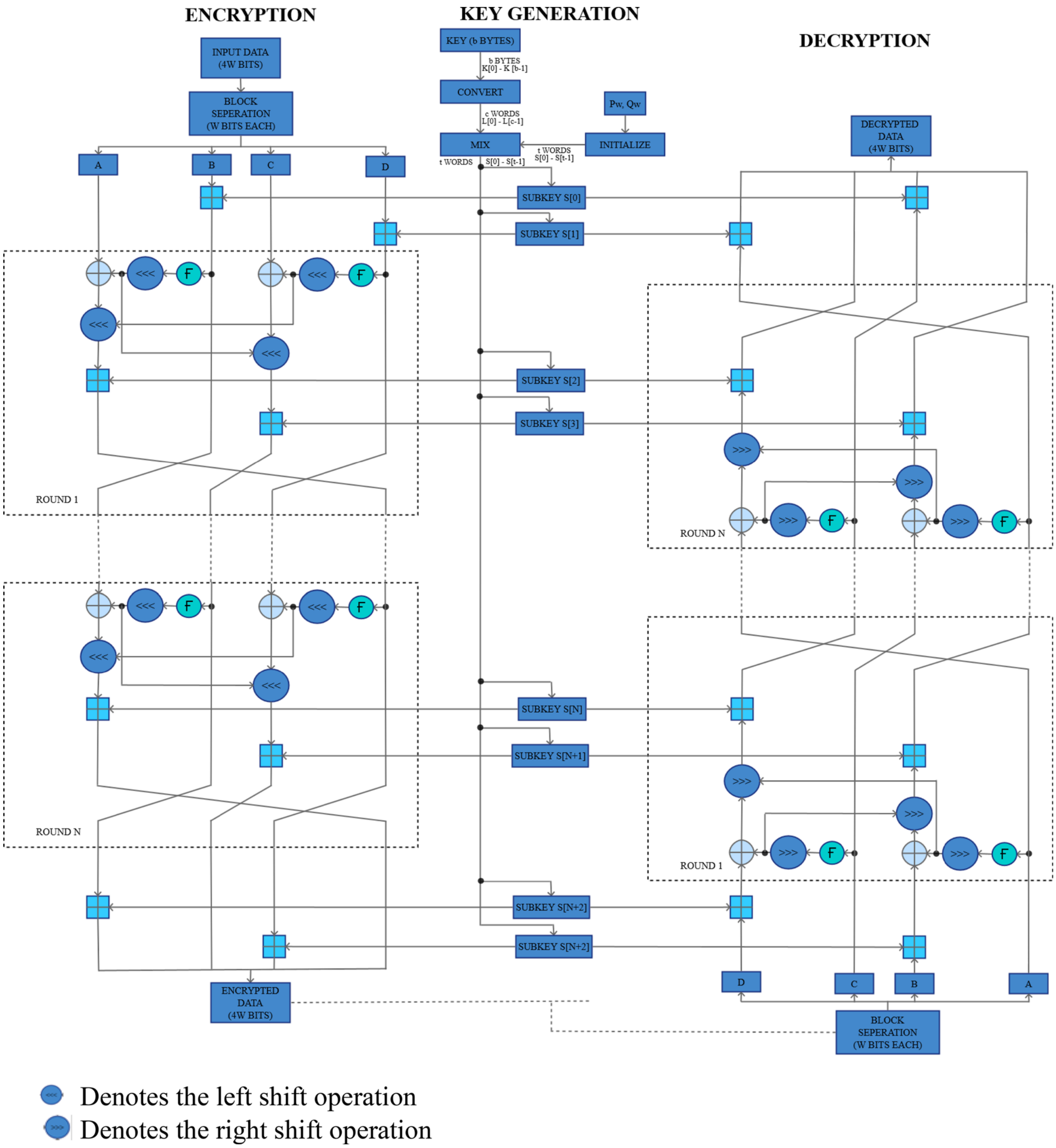
Fig. 8 Block diagram of RC5. ⊟ The subtractor



algorithms are comparatively secure. Figure 11 shows the block diagram of AES.

2.11 Scheme Based on Intertwining Chaotic Maps

Sam et al. [28] proposed a technique very similar to transformed logic maps using intertwining chaotic



Denotes the left shift operation

Denotes the right shift operation

Fig. 9 Block diagram of RC6. The left shift operation. The right shift operation

maps. Similar to the previously discussed algorithm, this technique can also be used for grayscale and colored images. The algorithm uses a total of nine keys (six random secret keys and three chaotic keys for permutation), XORing and channel mixing operations. Figure 12 shows the block diagram of scheme based on intertwining chaotic maps.

The plain image is stored in a two-dimensional array of $\{R_{i,j}, G_{i,j}, B_{i,j}\}$ pixels. In this, $1 \leq i \leq H$ and $1 \leq j \leq W$, where H and W represent height and width of the plain image in pixels.

Keys are generated using following mathematical expression where initial values that are $X_{1,1}$, $Y_{1,1}$ and $Z_{1,1}$ are random secret float values entered by the user. The μ values is in between 3.567 and 3.999 to achieve chaotic behavior and three float numbers (multipliers k1, k2, k3) are used to increase the randomness and uniform distribution of the key values

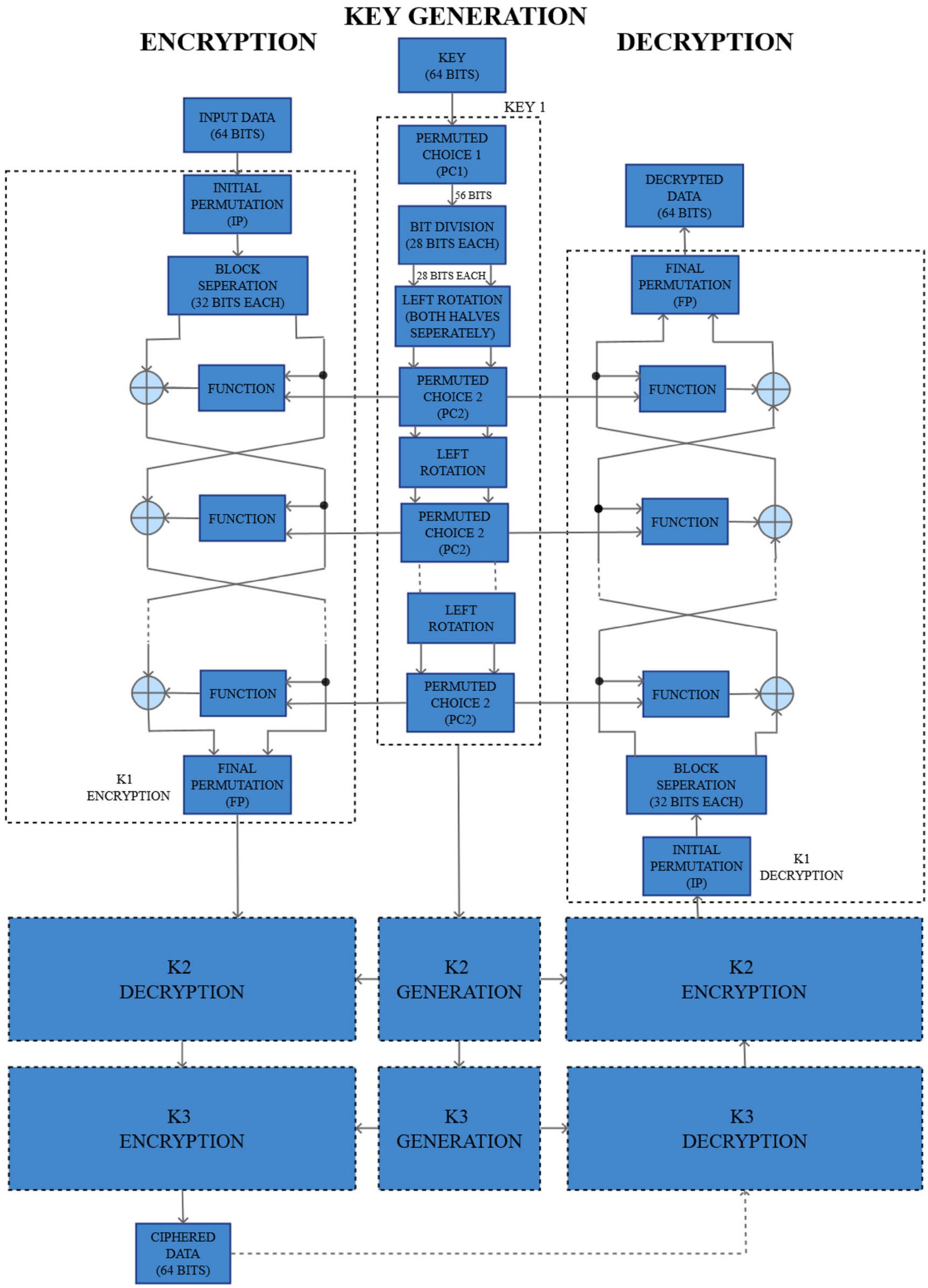


Fig. 10 Block diagram of Triple Data Encryption Standard

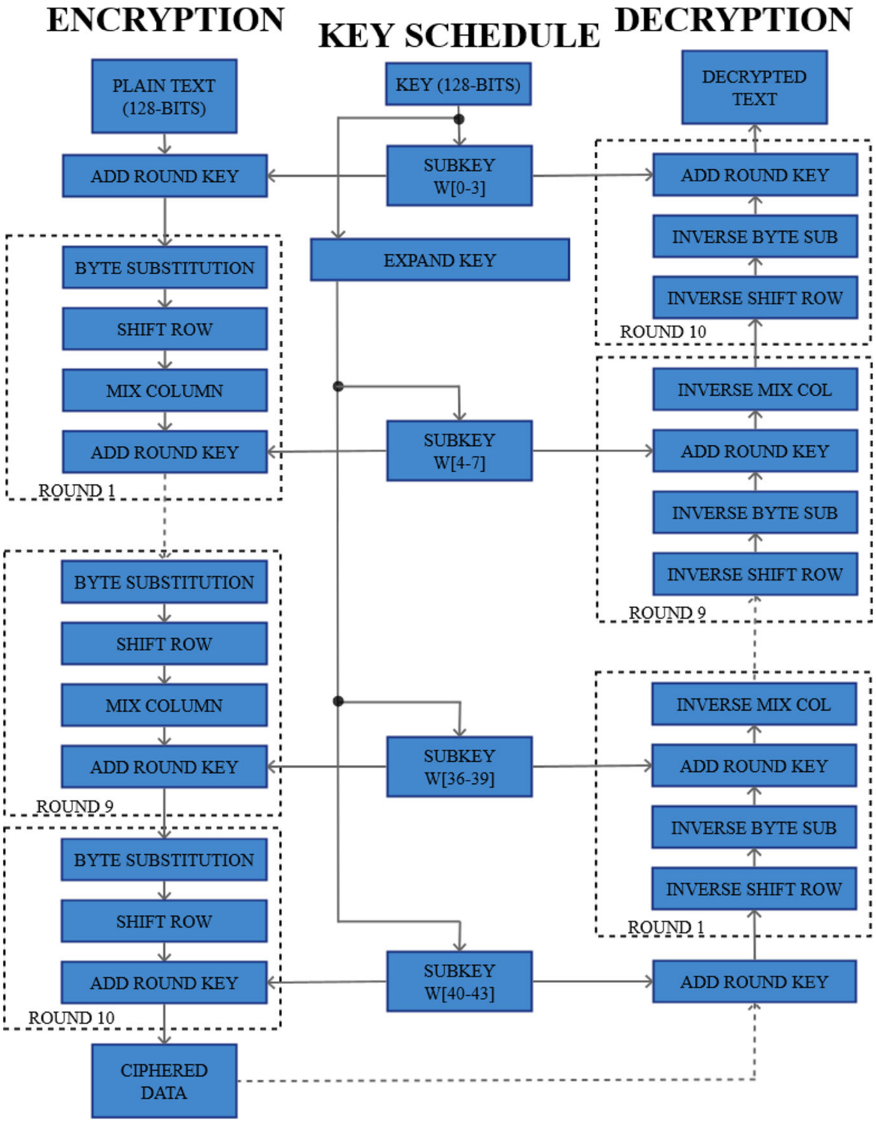
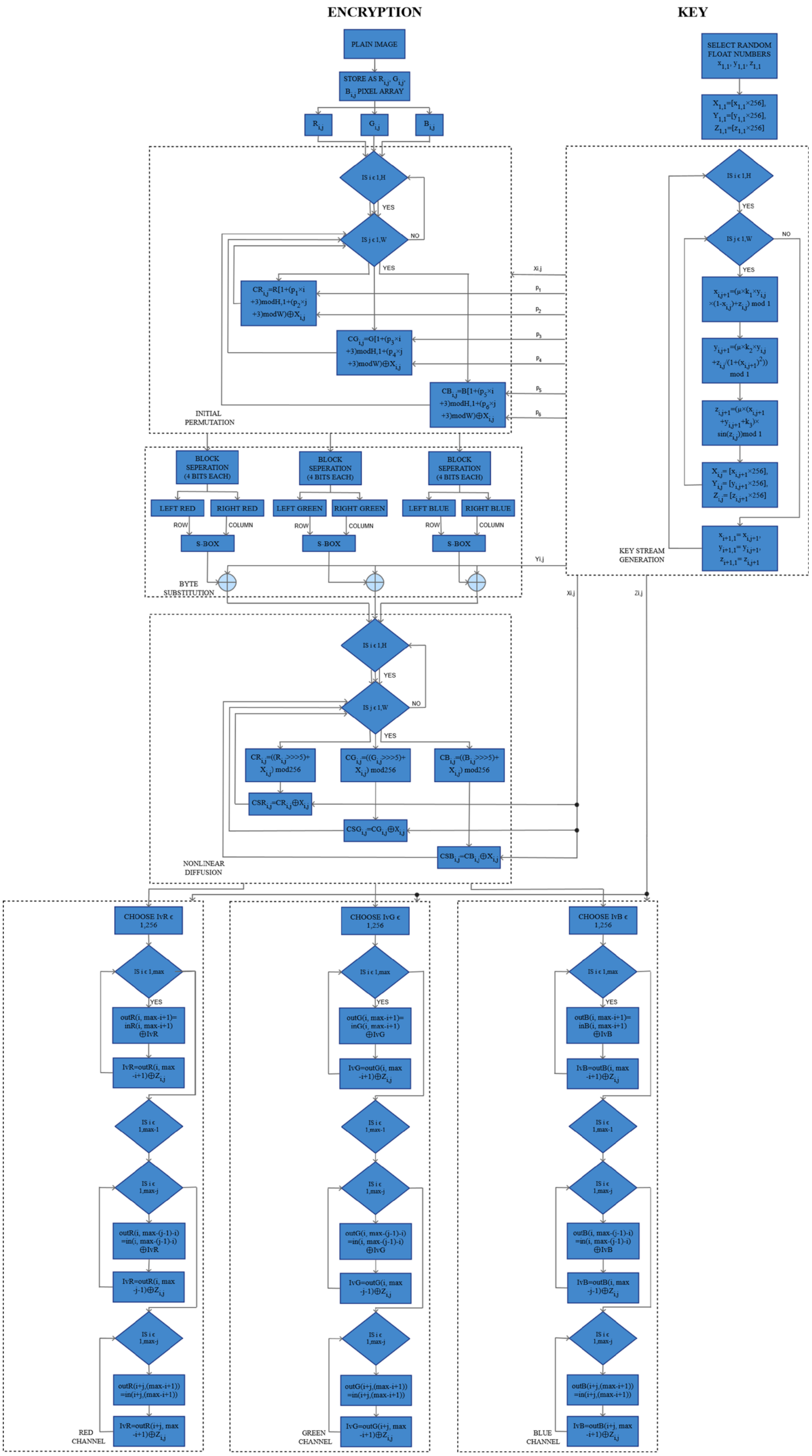


Fig. 11 Block diagram of Advanced Encryption Standard

Fig. 12 Block diagram of scheme based on intertwining chaotic maps


ENCRYPTION
KEY
PLAIN IMAGE
STORE AS $R_{i,j}$, $G_{i,j}$, $B_{i,j}$ PIXEL ARRAY
$R_{i,j}$
$G_{i,j}$
$B_{i,j}$
IS i ϵ 1,H
YES
IS j ϵ 1,W
NO
YES
$CR_{i,j}=R[1+(p_1\times i+3)\mathrm{mod}H,1+(p_2\times j+3)\mathrm{mod}W)\oplus X_{i,j}$
$CG_{i,j}=G[1+(p_3\times i+3)\mathrm{mod}H,1+(p_4\times j+3)\mathrm{mod}W)\oplus X_{i,j}$
$CB_{i,j}=B[1+(p_5\times i+3)\mathrm{mod}H,1+(p_6\times j+3)\mathrm{mod}W)\oplus X_{i,j}$
Xi,j
p_1
p_2
p_3
p_4
p_5
p_6
INITIAL PERMUTATION
SELECT RANDOM FLOAT NUMBERS $x_{1,1}$, $y_{1,1}$, $z_{1,1}$
$X_{1,1}=[x_{1,1}\times 256]$, $Y_{1,1}=[y_{1,1}\times 256]$, $Z_{1,1}=[z_{1,1}\times 256]$
IS i ϵ 1,H
YES
IS j ϵ 1,W
NO
YES
$x_{i,j+1}=(\mu\times k_1\times y_{i,j}\times(1-x_{i,j})+z_{i,j})$ mod 1
$y_{i,j+1}=(\mu\times k_2\times y_{i,j}+z_{i,j}/(1+(x_{i,j+1})^2))$ mod 1
$z_{i,j+1}=(\mu\times(x_{i,j+1}+y_{i,j+1}+k_3)\times\sin(z_{i,j}))$mod 1
$X_{i,j}=[x_{i,j+1}\times 256]$, $Y_{i,j}=[y_{i,j+1}\times 256]$, $Z_{i,j}=[z_{i,j+1}\times 256]$
$x_{i+1,1}=x_{i,j+1}$, $y_{i+1,1}=y_{i,j+1}$, $z_{i+1,1}=z_{i,j+1}$
KEY STREAM GENERATION
BLOCK SEPERATION (4 BITS EACH)
BLOCK SEPERATION (4 BITS EACH)
BLOCK SEPERATION (4 BITS EACH)
LEFT RED
RIGHT RED
LEFT GREEN
RIGHT GREEN
LEFT BLUE
RIGHT BLUE
ROW
COLUMN
ROW
COLUMN
ROW
COLUMN
S-BOX
S-BOX
S-BOX
Yi,j
BYTE SUBSTITUTION
Xi,j
Zi,j
IS i ϵ 1,H
YES
IS j ϵ 1,W
NO
YES
$CR_{i,j}=((R_{i,j}>>>5)+X_{i,j})$ mod256
$CG_{i,j}=((G_{i,j}>>>5)+X_{i,j})$ mod256
$CB_{i,j}=((B_{i,j}>>>5)+X_{i,j})$ mod256
$CSR_{i,j}=CR_{i,j}\oplus X_{i,j}$
$CSG_{i,j}=CG_{i,j}\oplus X_{i,j}$
$CSB_{i,j}=CB_{i,j}\oplus X_{i,j}$
NONLINEAR DIFFUSION
CHOOSE IvR ϵ 1,256
IS i ϵ 1,max
YES
outR(i, max-i+1)= inR(i, max-i+1) ⊕IvR
IvR=outR(i, max -i+1)⊕$Z_{i,j}$
IS i ϵ 1,max-1
IS i ϵ 1,max-j
outR(i, max-(j-1)-i) =in(i, max-(j-1)-i) ⊕IvR
IvR=outR(i, max -j-1)⊕$Z_{i,j}$
IS i ϵ 1,max-j
outR(i+j,(max-i+1)) =in(i+j,(max-i+1))
IvR=outR(i+j, max -i+1)⊕$Z_{i,j}$
RED CHANNEL
CHOOSE IvG ϵ 1,256
IS i ϵ 1,max
YES
outG(i, max-i+1)= inG(i, max-i+1) ⊕IvG
IvG=outG(i, max -i+1)⊕$Z_{i,j}$
IS i ϵ 1,max-1
IS i ϵ 1,max-j
outG(i, max-(j-1)-i) =in(i, max-(j-1)-i) ⊕IvG
IvG=outG(i, max -j-1)⊕$Z_{i,j}$
IS i ϵ 1,max-j
outG(i+j,(max-i+1)) =in(i+j,(max-i+1))
IvG=outG(i+j, max -i+1)⊕$Z_{i,j}$
GREEN CHANNEL
CHOOSE IvB ϵ 1,256
IS i ϵ 1,max
YES
outB(i, max-i+1)= inB(i, max-i+1) ⊕IvB
IvB=outB(i, max -i+1)⊕$Z_{i,j}$
IS i ϵ 1,max-1
IS i ϵ 1,max-j
outB(i, max-(j-1)-i) =in(i, max-(j-1)-i) ⊕IvB
IvB=outB(i, max -j-1)⊕$Z_{i,j}$
IS i ϵ 1,max-j
outB(i+j,(max-i+1)) =in(i+j,(max-i+1))
IvB=outB(i+j, max -i+1)⊕$Z_{i,j}$
BLUE CHANNEL

```
for i = 1 to 256
    for j = 1 to 256
        $x_{i,j+1} = \mu\times k1 \times y_{i,j} \times(1-x_{i,j})+z_{i,j}$
        $y_{i,j+1} = \mu\times k2 \times y_{i,j} +z_{i,j} \times 1/(1+(x_{i,j}+1)^{2})$ mod1
        $z_{i,j+1}= \mu\times(x_{i,j}+1+y_{i,j}+1+k3) \times \sin(z_{i,j})$ mod 1
        $X_{i,j}= x_{i,j+1}\times 256$
        $Y_{i,j}= y_{i,j+1}\times 256$
        $Z_{i,j}= z_{i,j+1}\times 256$
    end
    $x_{i+1,1} = x_{i,\, j+1}$
    $y_{i+1,1} = y_{i,\, j+1}$
    $z_{i+1,1} = z_{i,\, j+1}$
end
```

This process generates a matrix of chaotic values used for encryption process. Initial permutation is used to introduce confusion effect. Scheme permutes the pixels in the image, without changing its value.

```
for i = 1 to H
    for j = 1 to W
        $CR_{i,j} =R[1+(p1\times i+3)modH,1+(p2 \times j +3)modW] \oplus X_{i,j}$
        $CG_{i,j} =G[1+(p3\times i+3)modH,1+(p4 \times j +3)modW] \oplus X_{i,j}$
        $CB_{i,j} =B[1+(p3\times i+3)modH,1+(p4 \times j +3)modW] \oplus X_{i,j}$
    end

end
```

where $X_{i,j}$ is the first chaotic key, R[i, j], G[i, j], B[i, j] represent the red, green, blue channels in the plain-image and $CR_{i,j}$, $CG_{i,j}$, $CB_{i,j}$ denote the (i, j)th pixel of the permuted image. The method uses p1, p2, p3, p4, p5, p6 as odd random values for scrambling the image. In order to get a reversible permutation, p1,p3,p5 must be chosen so that to relatively prime to H and similarly p2, p4, p6 are relatively prime to W. It should be noted when image height and width are even numbers, the values p1, p2, p3, p4, p5, p6 must be odd.

In Byte Substitution, each individual RGB pixel byte of the state is replaced with a new byte by using the S-box of the AES (Advanced Encryption Standard) algorithm. The size of the S-box is 16 × 16. Here, the RGB component of each pixel is divided into two parts; the left half consists of the left most four bits and the right half consists of the right most four bits. They are denoted by LR, RR, LG, RG, LB, RB. For the red channel, the byte substitution is performed treating LR as the row number and RR as the column number of the S-box. Similarly, the other channel substitutions are done. The resultant values are XORed with the second chaotic key.

Nonlinear diffusion is obtained by the 5 Least Significant Bits (LSB) circular shift method. The resultant values are again XORed with the first chaotic key to all the red, green, and blue channels using mathematical formula described in flow chart where $X_{i,j}$ is the first chaotic key, $CR_{i,j}$ $CG_{i,j}$, $CB_{i,j}$ denotes the resultant values of the circular shift operation and

CSR$_{i,j}$, *CSG*$_{i,j}$, *CSB*$_{i,j}$ de notes the non linear diffusion image of the XORed operation

Next step is Sub-diagonal Diffusion which is obtained with the help of sub-diagonal XORing with the third chaotic key. Process is shown in flow chart where max the maximum size of the image, *inR* is and *outR* are the input and output of the image and *IvR*, *IvG*, *IvB* are the initial vector of each channel which may also be treated as an 8-bit secret key. $Z_{i,j}$ is the third chaotic key. The pixel is modified by XORing the first and second pixels with the chaotic key, the third pixel is modified by XORing modified second and third pixels with key and the process continues until the end of the image. The similar procedure is applied for the other channels. The algorithm has also shown a good coherence in analysis parameters such as key space analysis, differential analysis and statistical analysis.

2.12 Scheme Based on Chaotic Function Using Linear Congruence's

François et al. [29] presented a symmetric key image encryption algorithm with the aim to have large key spaces. The algorithm is developed to target the correlation present between the neighboring pixels. The key spacing algorithm includes three rounds variables R_1, R_2 and R_3. The number of rounds for a maximally secured encryption/decryption process is obtained by taking the maximum of the three. The algorithm utilizes a linear congruence based chaotic function for encryption and decryption processes. During the encryption process, the image is initially transformed into a binary 1D vector, I_0^b (see flow chart). A pseudo-randomized seed g in {1,…,L} initiates the relation of recurrence given by below chaotic equation. This binary vector is used to find the component of initial binary vector ($I_0^b[i]$) where i is the current position in the vector I_0^b and constructs a second vector component $I_0^b[j]$ in a new chaotic position $j = i + 1 + X_{i+1}$ using below chaotic equation. The elements of the vector I_0^b are transformed to $I_0^b[i] = Z3$ and $I_0^b[j] = Z1$ with

$$Z_1 = I_0^b[i],$$

$$Z2 = I_0^b[j] = I_0^b[i + 1 + X_{i+1}],$$

$$Z3 = Z1 \oplus Z2,$$

The bits of the vector I_0^b are gathered per package of 3 × 8 to form the cipher-image I_1. That constitutes the steps for one round for the cipher algorithm using the seed g. A complete encryption scheme produces a cipher-image IR, where R is the total number of rounds used to encrypt the plain-image I_0.

Chaotic equation used:

$$X_{n+1} = \left[\left[\left[X_n^2\right] mod\ S\right] \times X_n + X_g\right] mod\ S$$

with the initial position $X_0 = g$ and $X_g = g^2$, the seed g in(1,…,L) and L being the binary size of the image I0 (e.g. for a 256 gray-level image, its binary size is $L = 8 \times N \times M$ and $L = 3 \times 8 \times N \times M$ for a RGB-color image).

The decryption algorithm works on similar grounds. The algorithm is proved to be strong against attacks like brute-force attack. Figure 13 shows the block diagram of scheme based on chaotic function using linear congruence's.

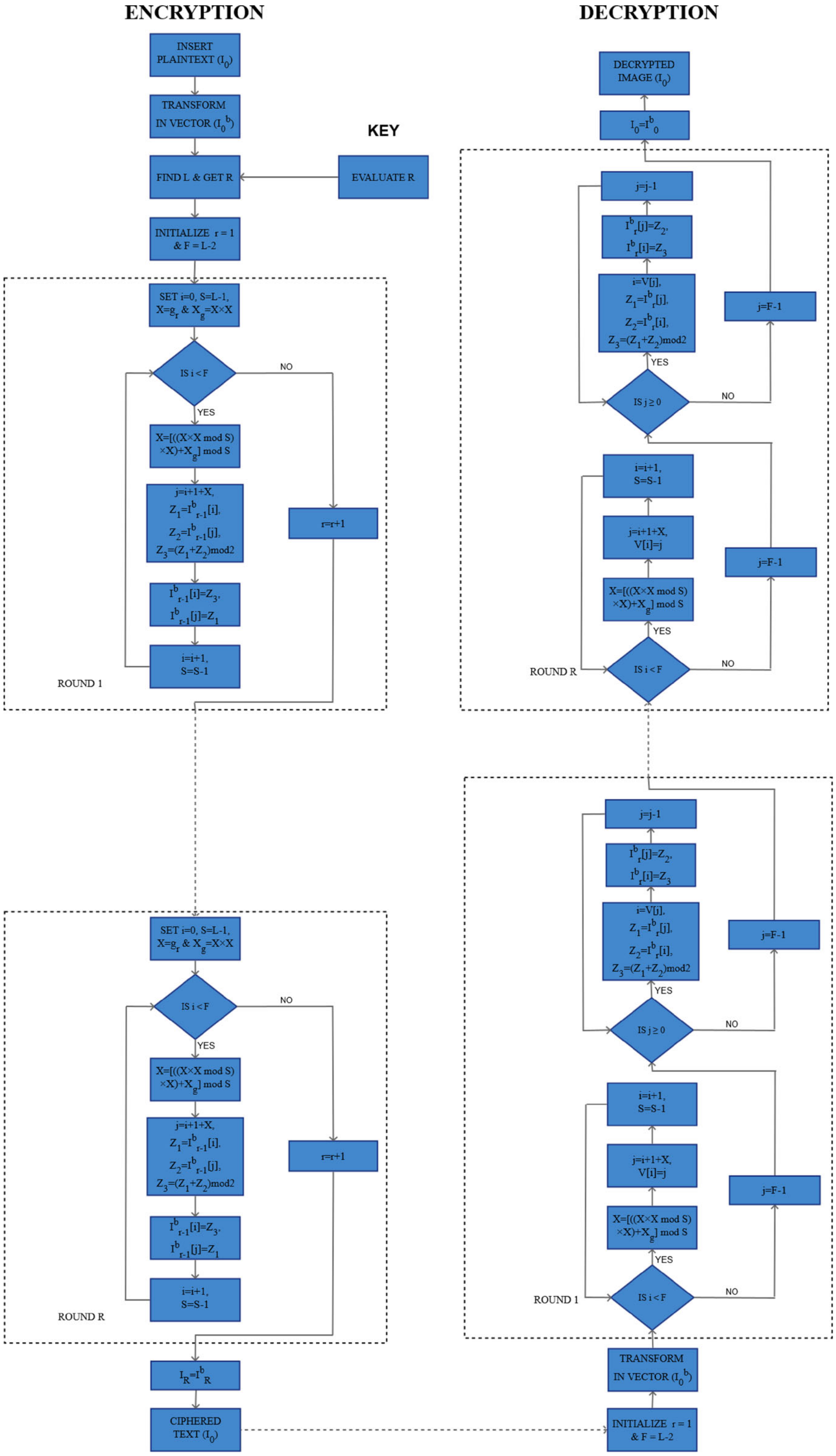
2.13 Scheme Based on Mixed Transform Logistic Maps

Sam et al. [30] presented a cryptosystem based on transformed logic maps and can be utilized for encryption of colored pixels. The algorithm uses a total of nine keys, six of which are denoted as odd secret keys while the rest three as chaotic keys, denoted as first, second and third chaotic maps. These maps are used to reduce vulnerability against known/chosen-plaintext attacks by performing non-linear diffusion, XORing and zig-zag processes respectively. The flow chart works as follows:

The plain image is stored in a two-dimensional array of {$R_{i,j}$, $G_{i,j}$, $B_{i,j}$} pixels $1 \le i \le H$ and $1 \le j \le W$, H and W represents the height and width of the plain image.

Keys are generated using following mathematical expression. $X_{1,1}$, $Y_{1,1}$ and $Z_{1,1}$ are random secret float values entered from the user. The μ values is in between 3.567 and 3.999 to achieve chaotic behavior and three float numbers (multipliers k1, k2, k3) are used increase the randomness and uniform distribution of the key values

Fig. 13 Block diagram of scheme based on chaotic function using linear congruences


ENCRYPTION
INSERT PLAINTEXT (I_0)
TRANSFORM IN VECTOR (I_0^b)
KEY
EVALUATE R
FIND L & GET R
INITIALIZE r = 1 & F = L-2
SET i=0, S=L-1, $X=g_r$ & X_g=X×X
IS i < F
NO
YES
X=[((X×X mod S) ×X)+X_g] mod S
j=i+1+X, $Z_1=I^b_{r-1}[i]$, $Z_2=I^b_{r-1}[j]$, $Z_3=(Z_1+Z_2)$mod2
r=r+1
$I^b_{r-1}[i]=Z_3$, $I^b_{r-1}[j]=Z_1$
i=i+1, S=S-1
ROUND 1
SET i=0, S=L-1, $X=g_r$ & X_g=X×X
IS i < F
NO
YES
X=[((X×X mod S) ×X)+X_g] mod S
j=i+1+X, $Z_1=I^b_{r-1}[i]$, $Z_2=I^b_{r-1}[j]$, $Z_3=(Z_1+Z_2)$mod2
r=r+1
$I^b_{r-1}[i]=Z_3$, $I^b_{r-1}[j]=Z_1$
i=i+1, S=S-1
ROUND R
$I_R=I^b_R$
CIPHERED TEXT (I_0)
DECRYPTION
DECRYPTED IMAGE (I_0)
$I_0=I^b_0$
j=j-1
$I^b_r[j]=Z_2$, $I^b_r[i]=Z_3$
i=V[j], $Z_1=I^b_r[j]$, $Z_2=I^b_r[i]$, $Z_3=(Z_1+Z_2)$mod2
j=F-1
YES
IS j ≥ 0
NO
i=i+1, S=S-1
j=i+1+X, V[i]=j
j=F-1
X=[((X×X mod S) ×X)+X_g] mod S
YES
ROUND R
IS i < F
NO
j=j-1
$I^b_r[j]=Z_2$, $I^b_r[i]=Z_3$
i=V[j], $Z_1=I^b_r[j]$, $Z_2=I^b_r[i]$, $Z_3=(Z_1+Z_2)$mod2
j=F-1
YES
IS j ≥ 0
NO
i=i+1, S=S-1
j=i+1+X, V[i]=j
j=F-1
X=[((X×X mod S) ×X)+X_g] mod S
YES
ROUND 1
IS i < F
NO
TRANSFORM IN VECTOR (I_0^b)
INITIALIZE r = 1 & F = L-2

```
for i = 1 to 256
    for j = 1 to 256
        $x_{i,j+1} = (3.735 \times (1+x_{i,j})^2 \times k_1 \times \sin(1/1+(y_{i,j})^2))\mathrm{mod}1$
        $y_{i,j+1} = (3.536 \times x_{i,j+1} \times k_2 \times \sin(x_{i,j+1} \times y_{i,j}) \times (1+(z_{i,j})^2))\mathrm{mod}1$
        $z_{i,j+1} = (3.828 \times x_{i,j+1} \times k_3 \times (1+y_{i,j+1} \times z_{i,j}))\mathrm{mod}1$
        $X_{i,j} = x_{i,j+1} \times 256$
        $Y_{i,j} = y_{i,j+1} \times 256$
        $Z_{i,j} = z_{i,j+1} \times 256$
    end
    $x_{i+1,1} = x_{i,j+1}$
    $y_{i+1,1} = y_{i,j+1}$
    $z_{i+1,1} = z_{i,j+1}$
end
```

This generates a matrix of chaotic values used in encryption process.

Initial permutation is used to introduce confusion effect. Scheme permutes the pixels in the image, without changing its value. The encryption process uses the 128-bit long secret key. There are six random odd integers keys (lying between 0 and 256) obtained from secret key. Then, the pixels are permuted using the operations specified in flow chart

The RGB diffusion is done by 4 bit circular shift method followed by addition between shifted value and the first chaotic key. The resultant value is xored with second chaotic key. The combination of 4 bit circular shift, secret key addition and xoring makes the encryption operation nonlinear and hence the system becomes strong against known/chosen plaintext attack. The procedure for the nonlinear diffusion is specified in flow chart, $X_{\mathrm{i,j}}$ and $Y_{\mathrm{i,j}}$ are the first and second chaotic key.

The last step is Zig-Zag diffusion. The diffusion is obtained with the help of zig-zag xoring and xoring with third chaotic key.

The cipher security is enhanced by introducing confusion effect obtained in a permutation stage and diffusion effect obtained in the pixel value diffusion stages. The algorithm has shown a good coherence in various analysis including key space analysis, differential analysis and statistical analysis. Figure 14 shows the block diagram of scheme based on mixed transform logistic maps.

2.14 Scheme Based on Peter De Jong Chaotic Map and RC4 Stream Cipher

Hanchinamani and Kulkarni [31] proposed a technique based on Peter De Jong chaotic map along with RC4 Stream Cipher. This encryption is based on three steps: permutation, pixel rotation, and diffusion. The permutation step scrambles the rows and columns followed by their alternate circular rotations. This is achieved by utilizing the chaotic maps. In the next step, all the pixel values are circularly rotated using M × N pseudo random numbers. Finally, a forward and backward diffusion to the pixel rotated image is applied.

Keyset of six values (X_0, Y_0, a, b, c, d) is fed into Peter De Jong Chaotic map to generate chaotic sequences which will be used in permutation, pixel rotation, and diffusion steps. In beginning, chaotic values generated are used as key for RC4 for generating pseudo random numbers. In first stage, permutation is performed in which the positions of rows and columns are scrambled along with induction of circular rotations in alternate orientations of rows and columns based on the position of chaotic values (based on PM′ and PN′ sequence) in their sorted form is proposed. In second stage every individual pixel is circularly rotated by using pseudo random numbers. In last stage, diffusion is performed twice in which image is scanned in two different ways: first, row-wise in alternative directions and then forward and backward diffusion is applied and after this it is scanned column-wise in same approach of alternative direction and then forward and backward diffusion is applied. The

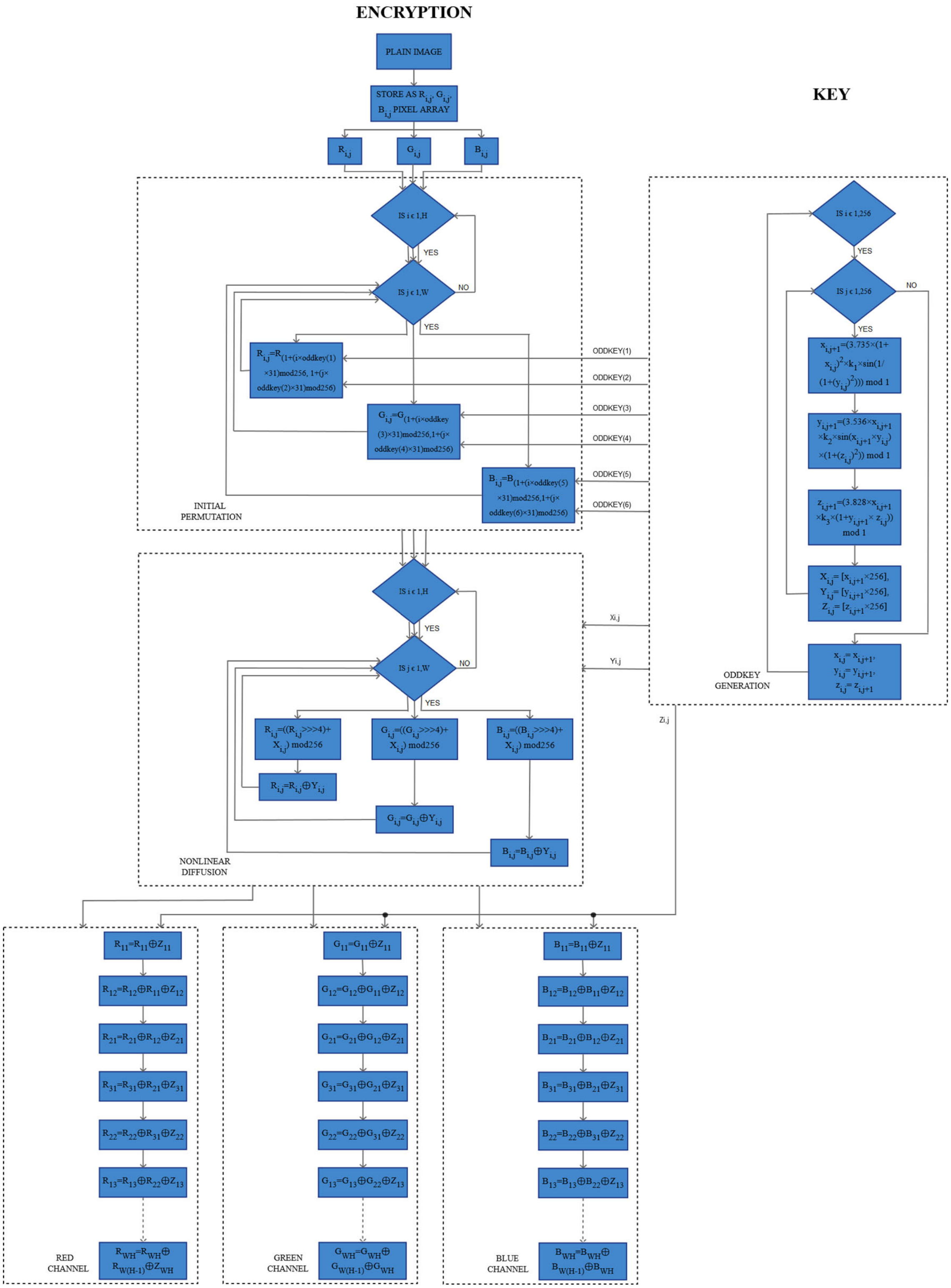

ENCRYPTION
KEY
PLAIN IMAGE
STORE AS $R_{i,j}$, $G_{i,j}$, $B_{i,j}$ PIXEL ARRAY
$R_{i,j}$
$G_{i,j}$
$B_{i,j}$
IS i ϵ 1,H
YES
IS j ϵ 1,W
NO
YES
$R_{i,j}=R_{(1+(i\times oddkey(1)\times 31)mod256,\ 1+(j\times oddkey(2)\times 31)mod256)}$
$G_{i,j}=G_{(1+(i\times oddkey(3)\times 31)mod256,1+(j\times oddkey(4)\times 31)mod256)}$
$B_{i,j}=B_{(1+(i\times oddkey(5)\times 31)mod256,1+(j\times oddkey(6)\times 31)mod256)}$
ODDKEY(1)
ODDKEY(2)
ODDKEY(3)
ODDKEY(4)
ODDKEY(5)
ODDKEY(6)
INITIAL PERMUTATION
IS i ϵ 1,256
YES
IS j ϵ 1,256
NO
YES
$x_{i,j+1}$=(3.735×(1+$x_{i,j}$)^2×k_1×sin(1/(1+($y_{i,j}$)^2))) mod 1
$y_{i,j+1}$=(3.536×$x_{i,j+1}$×k_2×sin($x_{i,j+1}$×$y_{i,j}$)×(1+($z_{i,j}$)^2)) mod 1
$z_{i,j+1}$=(3.828×$x_{i,j+1}$×k_3×(1+$y_{i,j+1}$× $z_{i,j}$)) mod 1
$X_{i,j}$= [$x_{i,j+1}$×256], $Y_{i,j}$= [$y_{i,j+1}$×256], $Z_{i,j}$= [$z_{i,j+1}$×256]
$x_{i,j}$= $x_{i,j+1}$, $y_{i,j}$= $y_{i,j+1}$, $z_{i,j}$= $z_{i,j+1}$
ODDKEY GENERATION
Xi,j
Yi,j
Zi,j
IS i ϵ 1,H
YES
IS j ϵ 1,W
NO
YES
$R_{i,j}$=(($R_{i,j}$>>>4)+$X_{i,j}$) mod256
$G_{i,j}$=(($G_{i,j}$>>>4)+$X_{i,j}$) mod256
$B_{i,j}$=(($B_{i,j}$>>>4)+$X_{i,j}$) mod256
$R_{i,j}$=$R_{i,j}$⊕$Y_{i,j}$
$G_{i,j}$=$G_{i,j}$⊕$Y_{i,j}$
$B_{i,j}$=$B_{i,j}$⊕$Y_{i,j}$
NONLINEAR DIFFUSION
R_{11}=R_{11}⊕Z_{11}
R_{12}=R_{12}⊕R_{11}⊕Z_{12}
R_{21}=R_{21}⊕R_{12}⊕Z_{21}
R_{31}=R_{31}⊕R_{21}⊕Z_{31}
R_{22}=R_{22}⊕R_{31}⊕Z_{22}
R_{13}=R_{13}⊕R_{22}⊕Z_{13}
R_{WH}=R_{WH}⊕$R_{W(H-1)}$⊕Z_{WH}
RED CHANNEL
G_{11}=G_{11}⊕Z_{11}
G_{12}=G_{12}⊕G_{11}⊕Z_{12}
G_{21}=G_{21}⊕G_{12}⊕Z_{21}
G_{31}=G_{31}⊕G_{21}⊕Z_{31}
G_{22}=G_{22}⊕G_{31}⊕Z_{22}
G_{13}=G_{13}⊕G_{22}⊕Z_{13}
G_{WH}=G_{WH}⊕$G_{W(H-1)}$⊕G_{WH}
GREEN CHANNEL
B_{11}=B_{11}⊕Z_{11}
B_{12}=B_{12}⊕B_{11}⊕Z_{12}
B_{21}=B_{21}⊕B_{12}⊕Z_{21}
B_{31}=B_{31}⊕B_{21}⊕Z_{31}
B_{22}=B_{22}⊕B_{31}⊕Z_{22}
B_{13}=B_{13}⊕B_{22}⊕Z_{13}
B_{WH}=B_{WH}⊕$B_{W(H-1)}$⊕B_{WH}
BLUE CHANNEL

Fig. 14 Block diagram of scheme based on mixed transform logistic maps

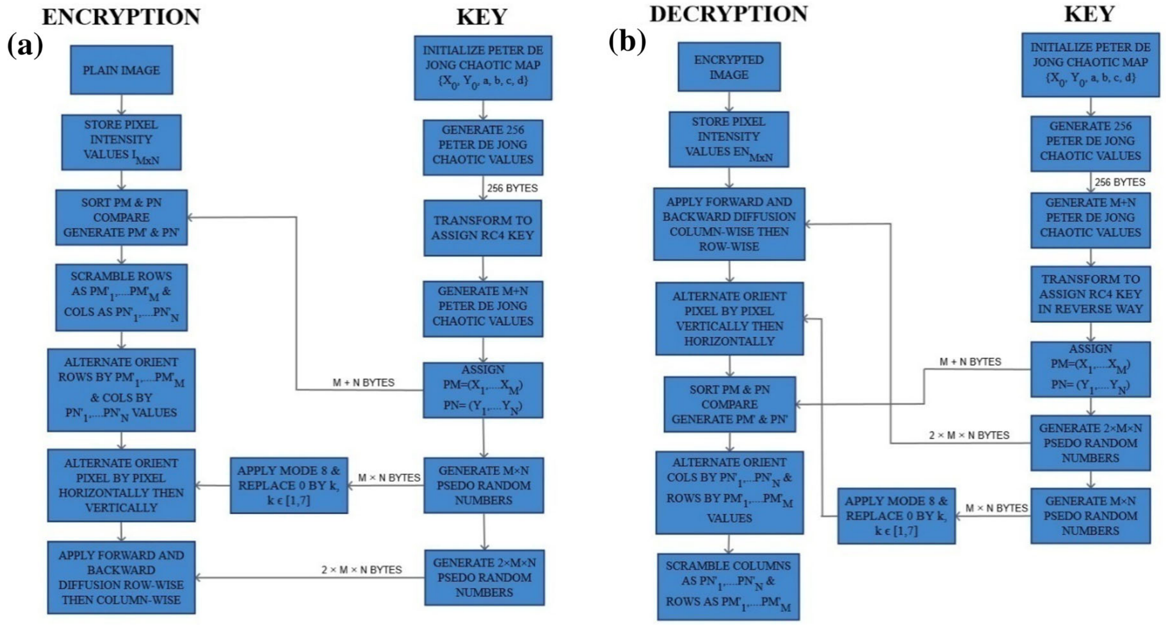

ENCRYPTION
(a)
KEY
PLAIN IMAGE
STORE PIXEL INTENSITY VALUES I_{MxN}
SORT PM & PN COMPARE GENERATE PM' & PN'
SCRAMBLE ROWS AS PM'_1,....PM'_M & COLS AS PN'_1,....PN'_N
ALTERNATE ORIENT ROWS BY PM'_1,....PM'_M & COLS BY PN'_1,....PN'_N VALUES
ALTERNATE ORIENT PIXEL BY PIXEL HORIZONTALLY THEN VERTICALLY
APPLY FORWARD AND BACKWARD DIFFUSION ROW-WISE THEN COLUMN-WISE
INITIALIZE PETER DE JONG CHAOTIC MAP {X_0, Y_0, a, b, c, d}
GENERATE 256 PETER DE JONG CHAOTIC VALUES
256 BYTES
TRANSFORM TO ASSIGN RC4 KEY
GENERATE M+N PETER DE JONG CHAOTIC VALUES
ASSIGN PM=(X_1,....X_M) PN= (Y_1,....Y_N)
M + N BYTES
GENERATE M×N PSEDO RANDOM NUMBERS
M × N BYTES
APPLY MODE 8 & REPLACE 0 BY k, k ϵ [1,7]
GENERATE 2×M×N PSEDO RANDOM NUMBERS
2 × M × N BYTES
DECRYPTION
(b)
KEY
ENCRYPTED IMAGE
STORE PIXEL INTENSITY VALUES EN_{MxN}
APPLY FORWARD AND BACKWARD DIFFUSION COLUMN-WISE THEN ROW-WISE
ALTERNATE ORIENT PIXEL BY PIXEL VERTICALLY THEN HORIZONTALLY
SORT PM & PN COMPARE GENERATE PM' & PN'
ALTERNATE ORIENT COLS BY PN'_1,....PN'_N & ROWS BY PM'_1,....PM'_M VALUES
SCRAMBLE COLUMNS AS PN'_1,....PN'_N & ROWS AS PM'_1,....PM'_M
INITIALIZE PETER DE JONG CHAOTIC MAP {X_0, Y_0, a, b, c, d}
GENERATE 256 PETER DE JONG CHAOTIC VALUES
256 BYTES
GENERATE M+N PETER DE JONG CHAOTIC VALUES
TRANSFORM TO ASSIGN RC4 KEY IN REVERSE WAY
ASSIGN PM=(X_1,....X_M) PN= (Y_1,....Y_N)
M + N BYTES
GENERATE 2×M×N PSEDO RANDOM NUMBERS
2 × M × N BYTES
GENERATE M×N PSEDO RANDOM NUMBERS
M × N BYTES
APPLY MODE 8 & REPLACE 0 BY k, k ϵ [1,7]

Fig. 15 **a** Encryption and **b** decryption of scheme based on mixed transform logistic maps

algorithm is so effective that a high level of security can be attained by using only two rounds of encryption and decryption processes. It is resilient against attacks like brute force, differential etc. Figure 15 shows the encryption and decryption of scheme based on mixed transform logistic maps.

2.15 Scheme Based on Chaotic Map and Vigenère Scheme

Bansal et al. [32] presented a cryptosystem in which the encryption algorithm utilizes a combination of chaotic maps and Vigenère scheme. This encryption algorithm is formed by two stages, namely diffusion and confusion. Further, the diffusion stage includes three steps: forward diffusion, matching process and backward diffusion. In order to increase the security of the encryption, the matching process is based on the Vigenère scheme. Next, in the confusion stage, the pixel positions are swapped by a position permutation process. The process is used for encryption of RGB images, where the red, green, and blue components of the image are processed similarly in isolated channels. The algorithm provides very low time complexity while providing comparable values of parameters like PSNR and Unified Average Changing Intensity (UACI) when compared to other chaotic schemes. Figure 16 shows the encryption scheme based on chaotic map and Vigenère scheme. The explanation of flow chart now follows:

Keys generation step involves generation of chaotic sequences using chaotic functions. These sequences will be used in subsequent steps of the encryption process. In Vigenère Scheme chaotic sequences are used to generate Vigenère matrices which are used in diffusion step. During Diffusion step input plain image is diffused using Forward diffusion, Vigenère and backward diffusion. Now after this step, a diffused image obtained passes through the Confusion step resulting in the encrypted image.

Chaotic maps use keyset (k1 to k7) to generate sequences which acts as keys. Two sequences are generated using Sine map, and K1 is initial value: sin_c, sin_d.In confusion process sin_c is used and sin_d is used while creating Vigenère table and diffusion process, both are $1 - d$ Vector. In sin_c, values range is between 0 and $M - 1$ and contains M elements, sin_d value's range is 0–255 and contains N elements (Size of image: $M \times N \times 3$). xlog_d is used for red diffusion and Vigenère table creation, while xlog_c is used for red confusion, for green, ylog_d will perform in the same way as that of xlog_d and ylog_c same as that of xlog_c, similarly zlog_d and zlog_c in case of Blue. Range of xlog_d& alike is 0–255, and

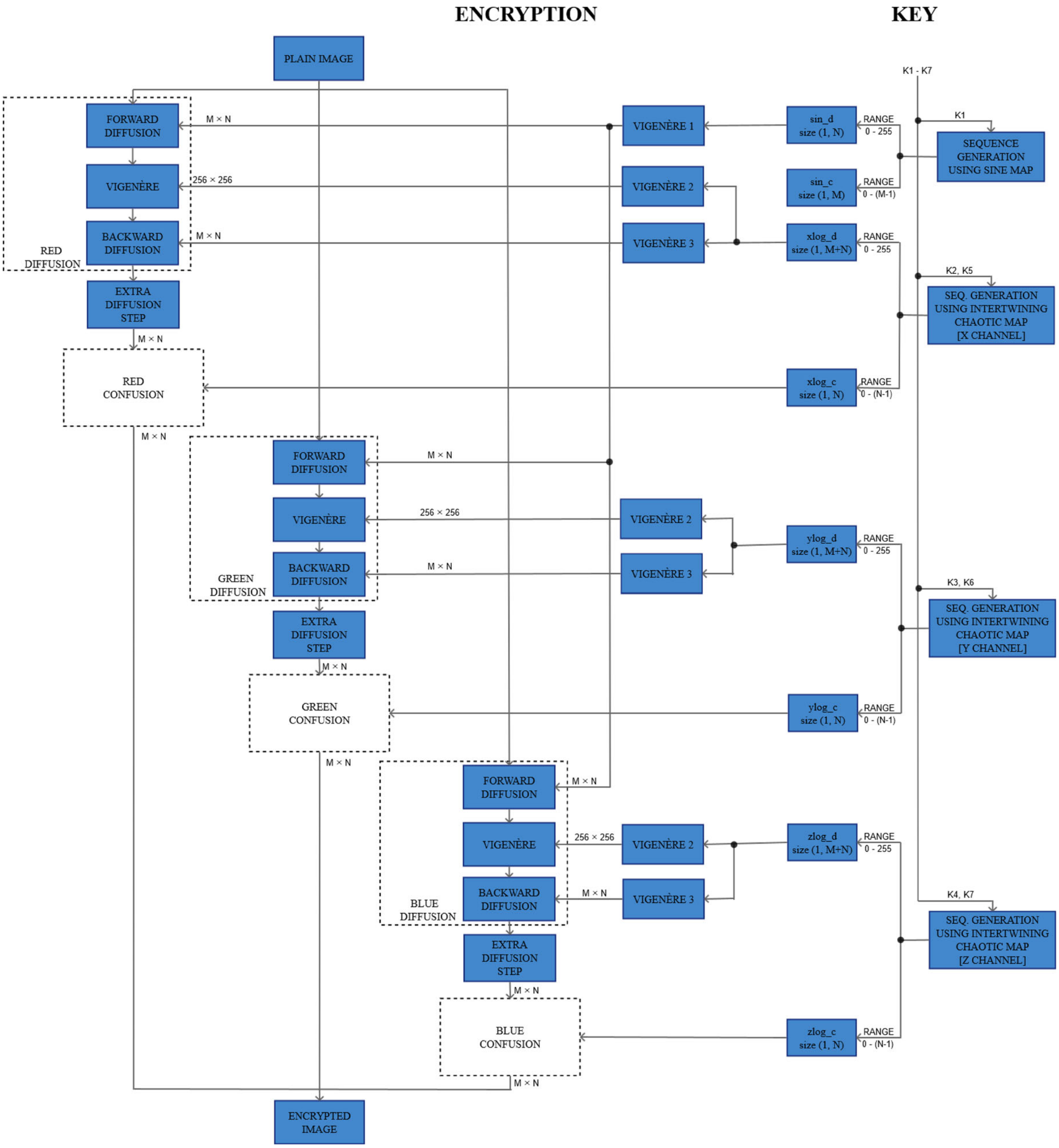


Fig. 16 Encryption scheme based on chaotic map and Vigenère scheme

contains M + N elements, & range of xlog_c and of similar sequences is 0 − (N − 1) and contains N elements. All sequences generated using Intertwining Logistic map are 1-day Vector.

3 Simulation Setup Parameters

3.1 Setup Parameters

A personal computer was used to conduct the simulations of current work. The image, machine and initial parameter specifications are provided in Table 1.

Table 1 Set up parameters

Specifications	
Processor	2.3 GHz Intel Core i5
Memory	8 GB DDR3 RAM
Operating system	Windows 10
Simulation platform	MATLAB
Version	2015
Size of images	128 × 128, 192 × 192, 256 × 256
Type	Color images
Key used in	
Vigenère, AES	2b7e151628aed2a6abf7158809cf4f3c
DES, blowfish	133457799bbcdff1
IDEA	5a14fb3e021c79e0608146a0117bff03
RC4	c3f4fc9088517fba6a2dea826151e7b22b7e151628aed2a6abf7158809cf4f3c
RC5	915f4619be41b2516355a50110a9ce91
RC6	de37a1fd8492d8efe714f1b7cc783aad
TDES	133457799bbcdff19bbcdff11334577933457799bbcdff11
Scheme 11	[33.1,37.3,35.7]
Scheme 12	[4713 654, 84 287, 7487, 1984, 12 314, 10, 74 120, 130 014, 95 210, 1914, 70 553, 2835, 19 800, 299 314, 83 721,610 990, 210, 65 521, 396, 1 109 094, 230 014, 63 010, 10 246]
Scheme 13 [k1, k2, k3]	[37.8, 39.8, 37.3]
[oddkey1, oddkey2, oddkey3, oddkey4, oddkey5, oddkey6]	[1, 5, 99, 111, 7, 77]
Scheme 14 [a, b, c, d, X0, Y0]	[1.77, 1.67, − 0.85, 2.1, 0.6, 0.4]
Scheme 15	[0.01,20,22,19,34,40,36]
Modified key	
Vigenère, Aes	2c7e151628aed2a6abf7158809cf4f3c
DES, blowfish	143457799bbcdff1
IDEA	5b14fb3e021c79e0608146a0117bff03
RC4	c4f4fc9088517fba6a2dea826151e7b22b7e151628aed2a6abf7158809cf4f3c
RC5	925f4619be41b2516355a50110a9ce91
RC6	df37a1fd8492d8efe714f1b7cc783aad
TDES	143457799bbcdff19bbcdff11334577933457799bbcdff11
Scheme 11	[33.11,37.3,35.7]
Scheme 12	[4714 654, 84 287, 7487, 1984, 12 314, 10, 74 120, 130 014, 95 210, 1914, 70 553, 2835, 19 800, 299 314, 83 721,610 990, 210, 65 521, 396, 1 109 094, 230 014, 63 010, 10 246]
Scheme 13 [k1, k2, k3]	[37.81, 39.8, 37.3]
Scheme 14 [a, b, c, d, X0, Y0]	[1.771, 1.67, − 0.85, 2.1, 0.6, 0.4]
Scheme 15	[0.011,20,22,19,34,40,36]

3.2 Performance Metrics Used

3.2.1 *Visual Assessment*

The three types of images were analyzed visually to determine if any information can be extracted by looking at the encrypted images.

3.2.2 *Statistical Analysis*

The encrypted images obtained by each algorithm are analyzed for any relation present among the pixels [33]. This was done by computing the histogram and correlation as described below.

1. Histogram analysis

Histogram of an image is a graphical representation of the frequency distribution of the pixel intensity values present in a digital image. Ideally, the histogram of an encrypted image should be spread uniformly and should have no similarity to the histogram of original image. A proper histogram distribution is required because many techniques, including AES, are at a risk of cryptanalysis using histograms [34, 35].

2. Correlation analysis

An image when encrypted should have no correlation between the adjacent pixels. Any correlation present can be used by an unauthorized user to recreate a part of the image, or worse the complete original image itself. The correlation coefficients range between − 1 and + 1, where the extremes shows a perfect negative or positive linear relation respectively. A coefficient value of zero represents no relation linear between the adjacent pixel values. In an image, the horizontal, vertical, and diagonal correlation coefficient between adjacent pixels can be given as follows:

$$r_{\alpha\beta} = \frac{cov(\alpha, \beta)}{\sqrt{D(\alpha)}\sqrt{D(\beta)}} \tag{1}$$

$$E(\alpha) = \frac{1}{N}\sum_{i=1}^{N}\alpha_i \tag{2}$$

$$D(\alpha) = \frac{1}{N}\sum_{i=1}^{N}(\alpha_i - E(\alpha))^2 \tag{3}$$

$$cov(\alpha, \beta) = \frac{1}{N}\sum_{i=1}^{N}(\alpha_i - E(\alpha))(\beta_i - E(\beta)) \tag{4}$$

where $cov(\alpha, \beta)$ is the covariance between original and encrypted image. $D(\alpha)$ is the variance of image. $E(\alpha)$ is the mean of the pixel values of the image

3.2.3 Differential Attack Analysis

Differential attack analysis are the tests performed to determine the changes in the encrypted image after providing a small change (generally single bit)in pixel or key value of the original image. In order to do this analysis, both the original image and the modified one are encrypted using the same encryption technique. One important parameter is the robustness of the encryption technique for which net pixel change ratio (NPCR) and unified average change in intensity (UACI) are utilized.

NPCR signifies the rate of change in number of pixels of the encrypted image when the original and pixel modified plain-images are compared. Let C1 and C2 be the cipher images for the original and pixel modified plain-images. NPCR is given as:

$$NPCR = \frac{\sum_{i=1}^{H}\sum_{j=1}^{W} D(i,j)}{W \times H} \times 100\% \tag{5}$$

where H and W are the height and width of the images. D is a bipolar array with size equivalent to the images C1 and C2. It has only 0 or 1 as components. D(i, j) is given as:

$$D(i,j) = \begin{cases} 0 & C1(i,j) = C2(i,j), \\ 1 & C1(i,j) \neq C2(i,j) \end{cases} \tag{6}$$

UACI is the difference in average intensity between the plain and encrypted image. It is given as:

$$UACI = \frac{1}{W \times H}\left[\sum_{i=1}^{H}\sum_{j=1}^{W}\frac{|C1(i,j) - C2(i,j)|}{2^L - 1}\right] \times 100\% \tag{7}$$

where L is the number of bits representing respective red, green, and blue channels.

3.2.4 Key Space Analysis

Key space analysis is an important parameter defining the feasibility of an encryption scheme to withstand a brute-force attack [24, 36]. In order to do this, the cipher should have a large combination of key spacing.

3.2.5 Key Sensitivity Analysis

Key sensitivity analysis deals with the effect of a small change in key used in the encryption algorithm on the encrypted image. The analysis is done by a pixel by pixel comparison of the two encrypted images and is observed by the NPCR for minute change (one bit change) in key value. An encryption scheme would be considered efficient based on two conditions: The first condition states that, if both the keys are used independently in the encryption to generate two separate encrypted images then both the encrypted images should be completely different. The second condition states that, if both the keys are used independently to decrypt the same encrypted image then only deciphering from the original key should provide the original image while the other should not provide any relevant result.

3.2.6 Quantitative Analysis: Image enhancement

The quantitative analysis is a comparison of image enhancement of the schemes. A higher image enhancement will produce a lesser distortion. PSNR and entropy metrics are used in this study and are defined below.

1. Peak signal-to-noise ratio (PSNR) analysis

It is the ratio between the maximum power component of the signal and the noise present in it. For its calculation, the plain-image is assumed to be the signal while the encrypted image is considered as the noise. Generally, a logarithmic decibel scale is used to describe PSNR as this type of scaling can be used for a compact representation of a wide range of signal. It is mathematically given as:

$$\mathrm{PSNR} = 20 \times \log_{10}\left(\frac{255}{\sqrt{\mathrm{MSE}}}\right)\mathrm{dB} \tag{8}$$

where MSE is the Mean Square Error and is a risk function i.e. the average of squares of the errors. MSE is given as:

$$MSE = \frac{1}{WH}\sum_{i=1}^{W}\sum_{j=1}^{H}[I(i,j) - K(i,j)]^2 \tag{9}$$

where I and K represents the pixel values of the original and the encrypted image and (i, j) represents the pixel location. W and H are dimensions of the image.

2. Information entropy analysis

Information entropy is a representation of the amount of randomness in given data. It directly relates to the distortion of the image. Mathematically, it is defined as

$$\mathrm{H(S)} = \sum_{i=0}^{n-1} P(S_i)\log_2\left(\frac{1}{P(S_i)}\right) \tag{10}$$

where n represents the total number of symbols, S_i is the pixel values and $P(S_i)$ represents the probability of occurrence of S_i. If the source is providing a total of 2^8 symbols, where each of $S = (S_1, S_2, S_3, \ldots, S_{255})$ has equal probability, then entropy H(S) = 8.

3.2.7 Time Complexity Analysis

Time complexity is the amount of time taken by a set of instructions to execute. Its manual approximation can be done by using the total elementary executable operations present in the set as the elementary operations have a fixed amount of time associated with them. Here, this time represents the time of encryption and decryption of the image and is computed by built-in operations. The time complexity depends on various factors like the system configuration and the image used. Hence, the parameters as mentioned in Table 1 were kept unchanged during the process.

4 Results

4.1 Visual Assessment

Table 2 presents the visual assessment of the encrypted images after applying the algorithms under investigation. It can be seen that the chaos-based techniques provides a high scrambling of the pixels of the original image in the encrypted image and no information about the original images can be visually extracted from the encrypted ones. Additionally, the encrypted images obtained by applying conventional techniques provide a significantly varying amount of scrambling. It can be observed that as the size of the original image increases, the amount of distortion in the encrypted image also increases. So, encryption by conventional techniques is visually more reliable for bigger sized images as compared to the smaller sized ones.

Some techniques show variations here too. Like, in visual cryptography technique the encrypted image is a randomly generated share and hence visually appears to be highly distorted irrespective of the image size. On the other hand, the encrypted image obtained by Vigenère scheme reveals a significant amount of information about the original image and

Table 2 Visual assessment and histogram analysis of the encryption schemes

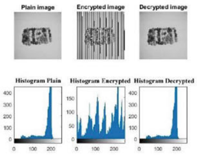
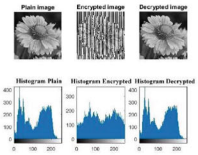
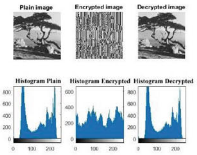
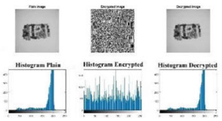
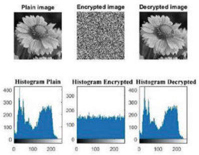
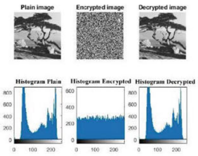
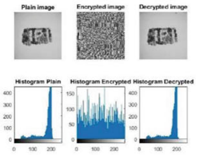
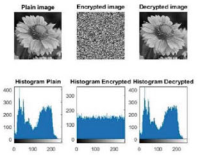
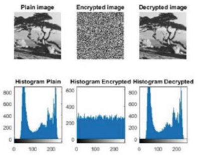
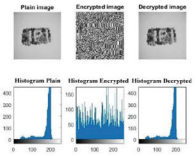
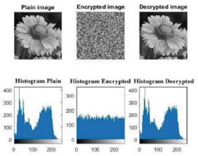
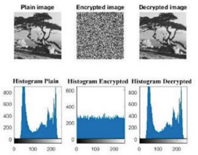
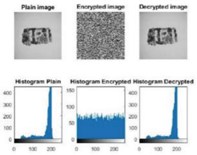
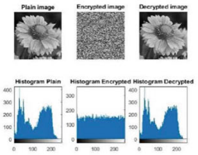
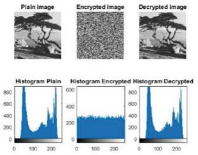
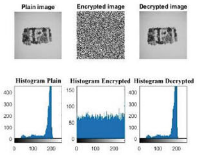
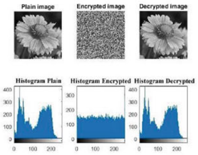
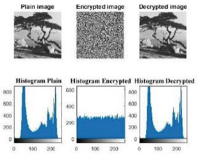
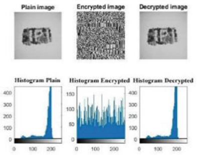
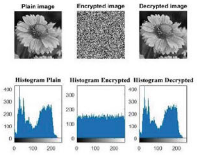
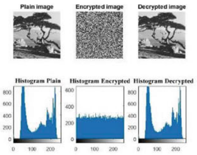
Size / Algorithm	128×128	192×192	256×256
Vigenère			
DES			
IDEA			
Blowfish			
Visual			
RC4			
RC5			

Table 2 continued

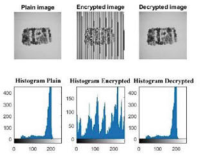
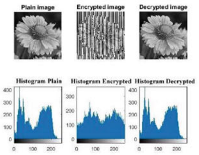
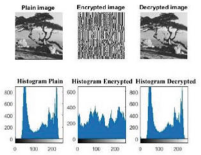
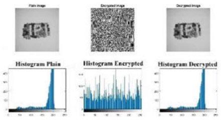
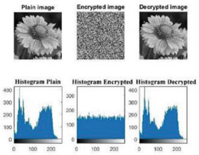
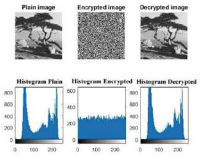
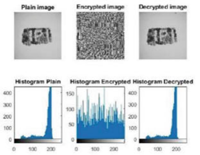
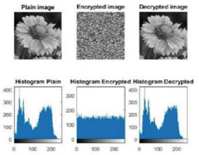
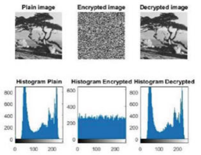
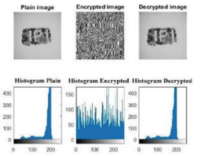
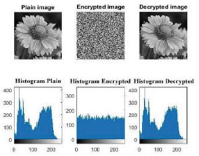
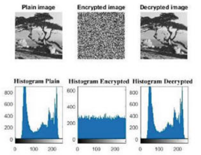
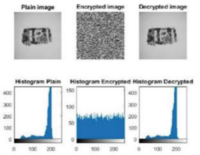
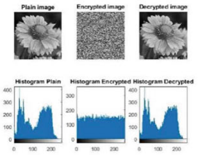
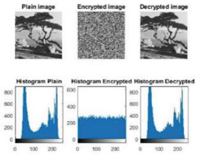
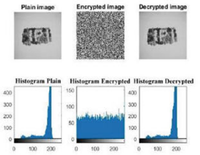
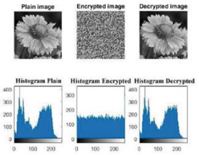
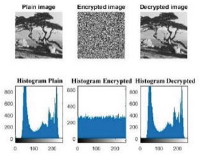
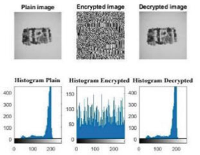
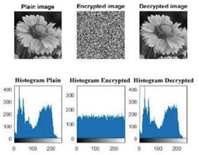
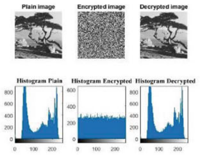
Size / Algorithm	128×128	192×192	256×256
Vigenère			
DES			
IDEA			
Blowfish			
Visual			
RC4			
RC5			

Table 3 Correlation coefficient of original images

Image	Horizontal	Vertical	Diagonal
128 × 128	0.955984	0.949743	0.913822
192 × 192	0.913524	0.938666	0.918481
256 × 256	0.940752	0.935892	0.926713

hence cannot be considered efficient on visual assessment grounds.

Finally, all the schemes provide the decrypted images similar to the original images which ensure the reliability of decrypted image if security is ensured.

4.2 Statistical Analysis

1. Histogram analysis

Table 2 also presents the histograms of original, encrypted and decrypted images. The histograms of images encrypted by chaos-based schemes are uniformly distributed and show no statistical resemblance with the histograms of original images. So, these techniques provide a strong resistance against the statistical attacks. Similar to the visual analysis, the histograms of encrypted images obtained by applying conventional techniques though show no resemblance with the histograms of original image but some of these have spikes and are not uniformly distributed. This decreases their resistance against the statistical attacks. The histograms are more uniformly distributed in some schemes like RC4 and visual but for many others the histograms of larger encrypted images are more uniformly distribute.

2. Correlation analysis

The analysis is performed by utilizing 10,000 random pixels pairs in the plain and encrypted images. Each of the pixel pairs contains one randomly selected pixel and a pixel adjacent to it. Table 3 presents the horizontal, vertical and diagonal correlation coefficients of the original images used.

Figures 17, 18 and 19 present the horizontal, vertical and diagonal correlation coefficients of original images and for the encrypted images obtained after applying various algorithms under investigation. It can be seen that all the chaos-based schemes and some conventional techniques like RC4 and visual provided very low correlation coefficient values for all three test images. It reflects the high resistance of these schemes against statistical attacks.

For the conventional encryption schemes left, most showed higher values for either horizontal or vertical correlation. This represents their reduced resistance against the statistical attacks. But these values are still comparatively lesser than the original correlation coefficients, hence ensuring security against statistical attacks up to some extent. A general trend of lower correlation can also be seen as the image size increases which support the histogram observation of increase in

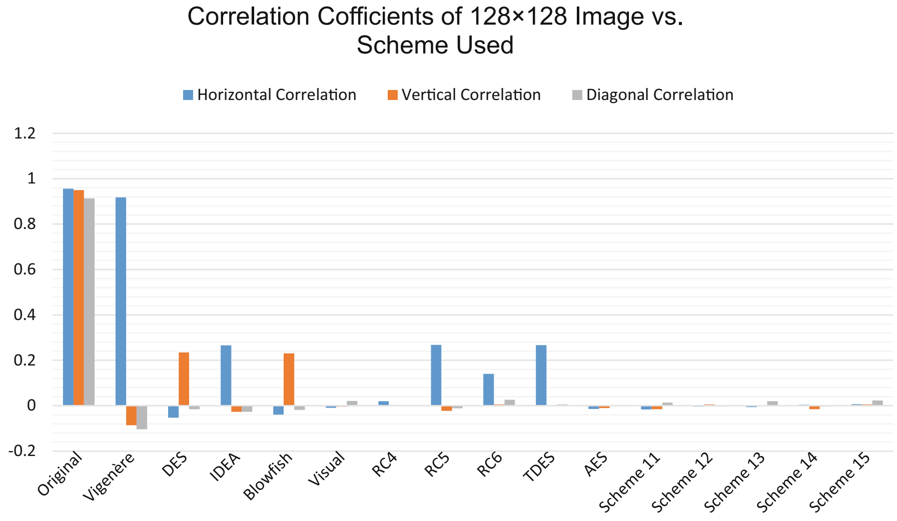


Fig. 17 Correlation coefficients of original and encrypted 128 × 128 images

Fig. 18 Correlation coefficients of original and encrypted 192 × 192 images

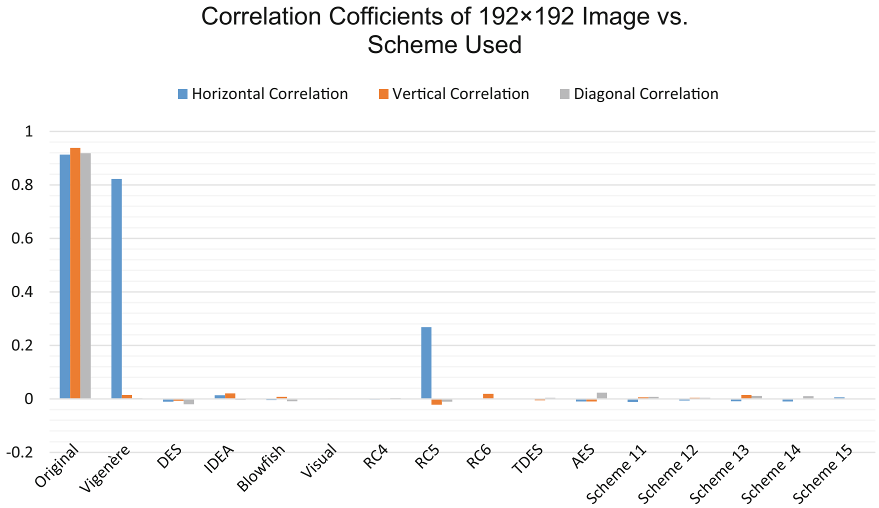
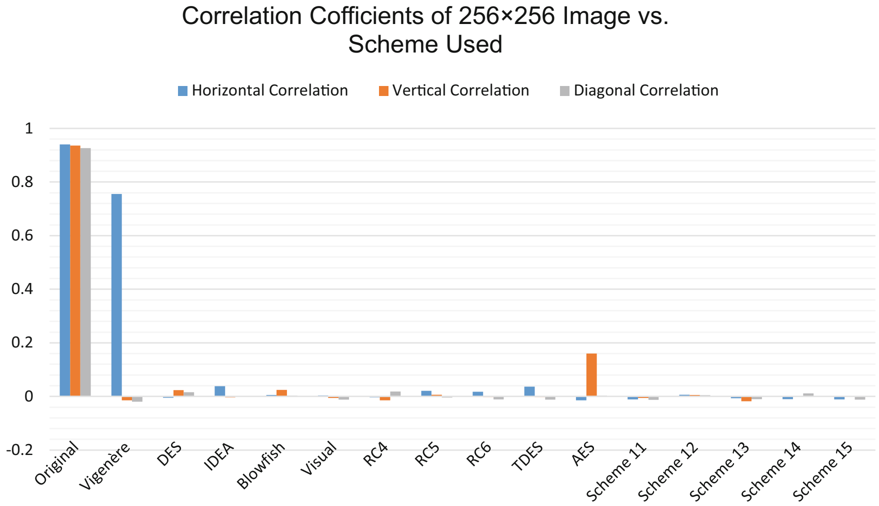


Fig. 19 Correlation coefficients of original and encrypted 256 × 256 images



uniform distribution with size. These indicate an increase in resistance against statistical attacks as the image size increases.

The correlation plots of the original and the encrypted images are provided in Table 4. It can be seen that the correlation plots of the original images are highly non-uniformly distributed. The plots are concentrated at the corners and sometimes along the central line too, but are scarcer in other regions of the graph. This is the original correlation distribution of test images in horizontal, vertical, and diagonal directions.

From all the encryption schemes used, the Vigenère scheme provided the encrypted images with maximum amount of correlation left. The correlation graphs for these images still show a significantly higher density along the central line. The graphs also contain high density patches which have no direct relation with the original correlation graphs, but these patches result indicate interlinked correlation and disobeys the even distribution property of an ideal correlation graph required to resist statistical attacks.

For rest of the conventional encryption schemes, the correlation graphs of the encrypted images are properly scattered and more uniformly distributed. Occasionally, central lines with high concentration are present for horizontal correlation in these graphs which shows a slight similarity with the original

Table 4 Correlation plots (horizontal, vertical and diagonal) of original and encrypted images

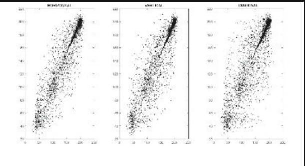
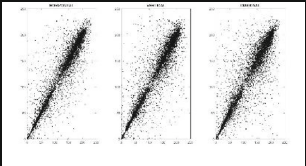
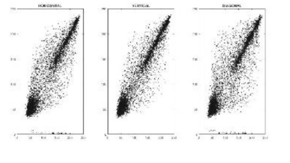
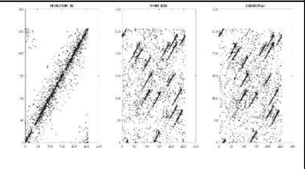
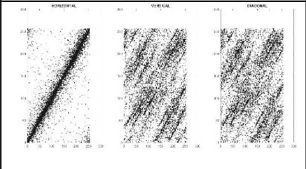
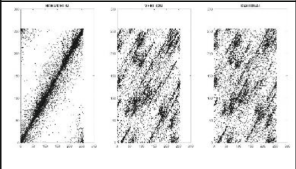
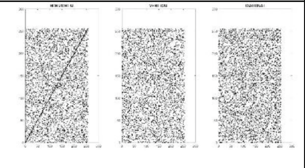
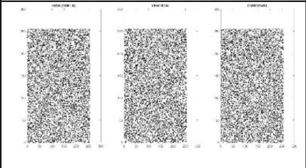
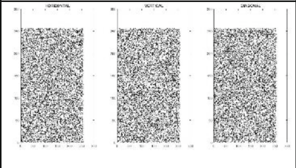
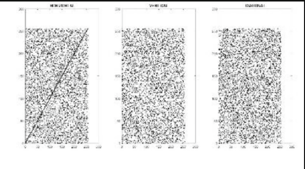
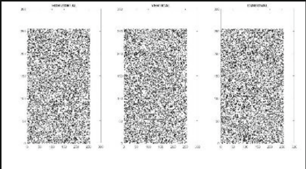
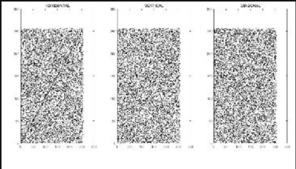
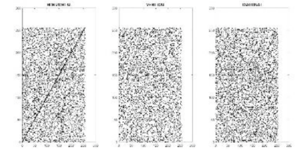
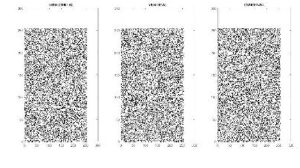
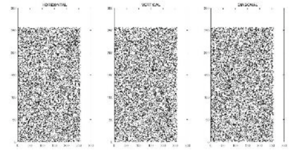
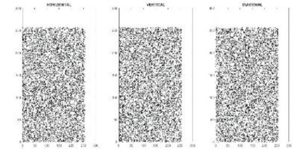
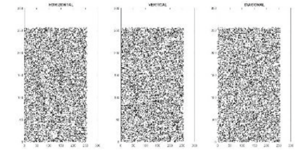
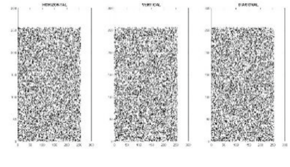
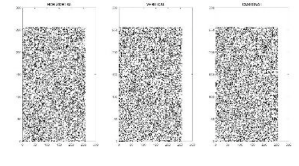
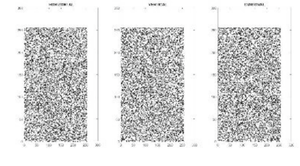
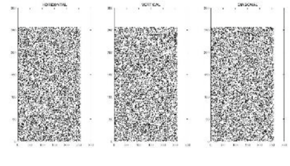
Size/ Algorithm	128×128	192×192	256×256
Original			
Vigenère			
DES			
IDEA			
Blowfish			
Visual			
RC4			

Table 4 continued

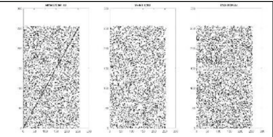
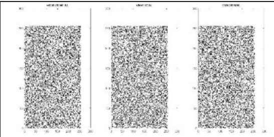
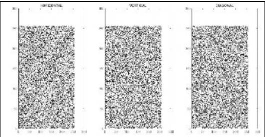
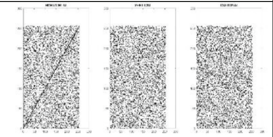
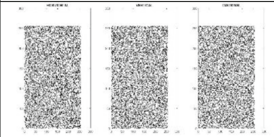
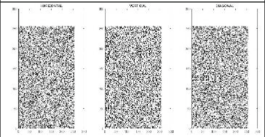
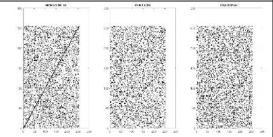
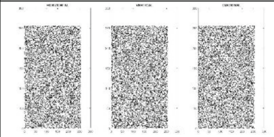
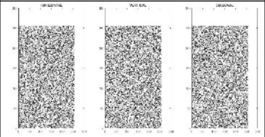
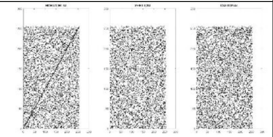
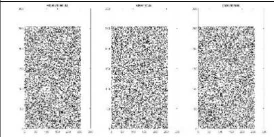
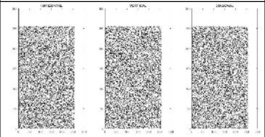
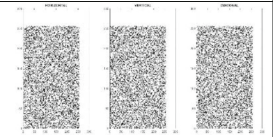
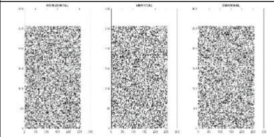
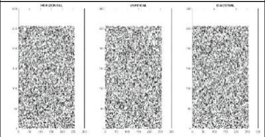
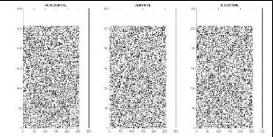
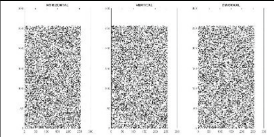
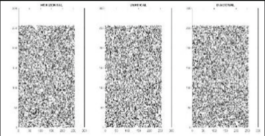
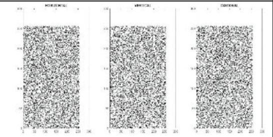
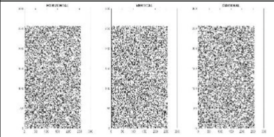
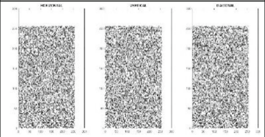
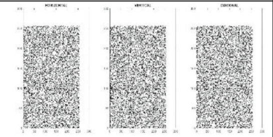
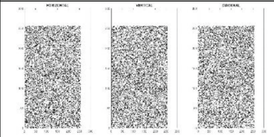
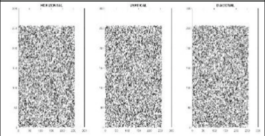
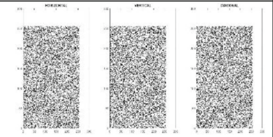
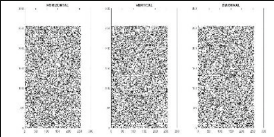
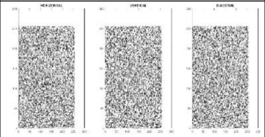
RC5			
RC6			
TDES			
AES			
Scheme 11			
Scheme 12			
Scheme 13			
Scheme 14			
Scheme 15			

image. Hence, these techniques are at some risk of the statistical attacks. For the chaotic encryption schemes, the correlation graphs are highly uniformly distributed and present no high density regions, which show their high resistance against the statistical attacks.

4.3 Differential Attack Analysis

The analysis is performed by observing NPCR and UACI obtained after pixel change and key change separately. Tables 5 and 6 present the NPCR and UACI values obtained for different images for respective one-bit change in pixel and key values for different algorithms under investigation. It can be seen that, for single pixel change in chaotic schemes, the NPCR and UACI values for all three test images are at more than 99.4 and 33.2% respectively. These values are very high and it is because of the diffusion stage present in these algorithms. The diffusion stage ensures a large change in the encrypted image even if a single pixel in the original image is changed, i.e. the diffusion stage makes the process highly sensitive to initial pixel configuration. This makes the chaotic schemes highly resistive against the differential attacks.

For single pixel change in conventional cryptography schemes, a significant lesser NPCR and UACI values are obtained. The schemes like Vigenère, Visual and RC4 provide the least NPCR and UACI values among the schemes used. This shows their vulnerability against the differential attacks. On the other hand the schemes like RC6 and AES showed the highest NPCR and UACI values among the conventional schemes. Both, the NPCR and UACI values, were higher for these schemes where the UACI values increases more than 1000 times than the previous mentioned schemes. Even then, these values are significantly lesser than the values obtained by the chaotic schemes. It clearly shows that these conventional schemes are not very much effective against the differential attacks. Moreover, as the image size increases, a decrease in the values can be observed indicating an increase in vulnerability with size.

High values of NPCR and UACI are one of the most important security criteria. Many researchers have used the vulnerability of algorithms providing lower values of these parameters for cryptanalysis [37, 38].

Table 5 NPCR and UACI values for one-bit change in pixel value

Size/algorithm	128 × 128		192 × 192		256 × 256	
Scheme	NPCR	UACI	NPCR	UACI	NPCR	UACI
Vigenère	6.10E − 05	4.79E − 06	2.71E − 05	2.13E − 06	1.53E − 05	1.20E − 06
DES	4.88E − 04	1.17E − 04	2.17E − 04	5.12E − 05	1.22E − 04	5.43E − 05
IDEA	4.88E − 04	1.25E − 04	2.17E − 04	8.83E − 05	1.22E − 04	4.09E − 05
Blowfish	4.88E − 04	1.48E − 04	2.17E − 04	7.65E − 05	1.22E − 04	2.74E − 05
Visual	6.10E − 05	2.39E − 07	2.71E − 05	1.06E − 07	1.53E − 05	5.98E − 08
RC4	6.10E − 05	2.87E − 06	2.71E − 05	1.28E − 06	1.53E − 05	7.18E − 07
RC5	4.88E − 04	1.82E − 04	2.17E − 04	4.27E − 05	1.22E − 04	4.58E − 05
RC6	9.77E − 04	2.31E − 04	4.34E − 04	1.49E − 04	2.44E − 04	9.07E − 05
TDES	4.88E − 04	1.57E − 04	2.17E − 04	6.45E − 05	1.22E − 04	4.97E − 05
AES	9.77E − 04	3.21E − 04	4.34E − 04	1.42E − 04	2.44E − 04	6.21E − 05
Scheme 11	9.94E − 01	3.32E − 01	9.96E − 01	3.33E − 01	9.93E − 01	3.33E − 01
Scheme 12	9.94E − 01	3.32E − 01	9.95E − 01	3.34E − 01	9.96E − 01	3.34E − 01
Scheme 13	9.95E − 01	3.33E − 01	9.96E − 01	3.33E − 01	9.93E − 01	3.33E − 01
Scheme 14	9.93E − 01	3.35E − 01	9.94E − 01	3.35E − 01	9.87E − 01	3.35E − 01
Scheme 15	9.96E − 01	3.34E − 01	9.96E − 01	3.35E − 01	9.96E − 01	3.35E − 01

Table 6 NPCR and UACI values for one-bit change in key value

Size/algorithm	128 × 128		192 × 192		256 × 256	
Scheme	NPCR	UACI	NPCR	UACI	NPCR	UACI
Vigenère	0.0625	2.45E − 04	0.0625	3.26E − 04	0.0625	6.40E − 04
DES	0.9974365	0.332803	0.9958767	0.333614	0.9968872	0.335559
IDEA	0.991699	3.34E − 01	0.996582	3.34E − 01	0.996826	3.35E − 01
Blowfish	0.9973145	0.339515	0.995497	0.333403	0.9963531	0.33368
Visual	6.10E − 05	2.39E − 07	2.71E − 05	3.19E − 07	1.53E − 05	5.98E − 08
RC4	0.9969482	0.33623	0.9964735	0.335016	0.9962463	0.335093
RC5	0.9945679	0.330237	0.9961209	0.33347	0.9956512	0.333082
RC6	0.9967651	0.337234	0.9957682	0.336613	0.9960327	0.334877
TDES	0.993713	0.337402	0.996012	0.334754	0.995987	0.01115
AES	0.9971313	3.37E − 01	0.9956326	0.333483	0.9958344	0.334545
Scheme 11	0.996582	0.332921	0.996121	0.33492	0.996145	0.335464
Scheme 12	0.995789	0.334284	0.995813	0.333934	0.996084	0.334213
Scheme 13	0.995911	0.333911	0.996247	0.335255	0.99649	0.33413
Scheme 14	0.996826	0.334371	0.996718	0.334069	0.99585	0.335553
Scheme 15	0.995859	0.335178	0.996257	0.33374	0.99617	0.335513

4.4 Key Space Analysis

Table 7 provides the key spaces of the algorithms under investigation. It can be clearly seen that all of the chaos-based schemes have a key space large enough to resist the brute force attacks. As some conventional schemes have smaller key spaces, they become vulnerable to this most basic type of attack. The schemes like Blowfish can utilize variable key size and the key space can be increase more than mentioned in the table by using a key of larger size.

Table 7 Key space analysis of algorithms under investigation

Algorithm	Key space
Vigenère	2^{128}
DES	2^{56}
IDEA	2^{128}
Blowfish	2^{64}
RC4	2^{256}
RC5	2^{128}
RC6	2^{128}
TDES	2^{168}
AES	2^{128}
Scheme 11	2^{216}
Scheme 12	2^{126}–2^{147}
Scheme 13	2^{192}
Scheme 14	2^{384}
Scheme 15	2^{448}

4.5 Key Sensitivity Analysis

Table 6 shows the NPCR and UACI value after one-bit key change which is a representation of key sensitivity of a presented algorithm. It shows that, the chaos-based schemes provide very high values of NPCR and UACI values of around 99.6 and 33.4% respectively. The high values indicate their high key sensitivity. As generation of a complex and efficient key was one of the primary concern of the creators of conventional cryptography schemes, very high values of NPCR and UACI can be seen in most of these schemes. These values are over 99 and 33% respectively with cases like Blowfish where these can be observed as high as 99.7 and 33.9% respectively.

It is very obvious to understand that as the key generation of visual is in form of an encrypted share and generally depends on a random function, hence the scheme will have a poor key sensitivity. Also for Vigenère, it is obvious to have a poor key sensitivity response as the keys generated in the scheme have a repeating nature and is one of the major drawbacks of

Fig. 20 PSNR values obtained for the three test images

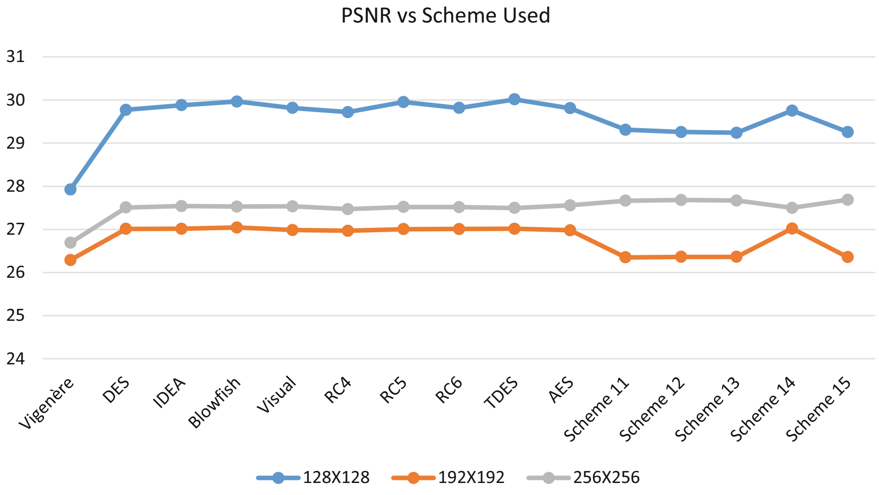
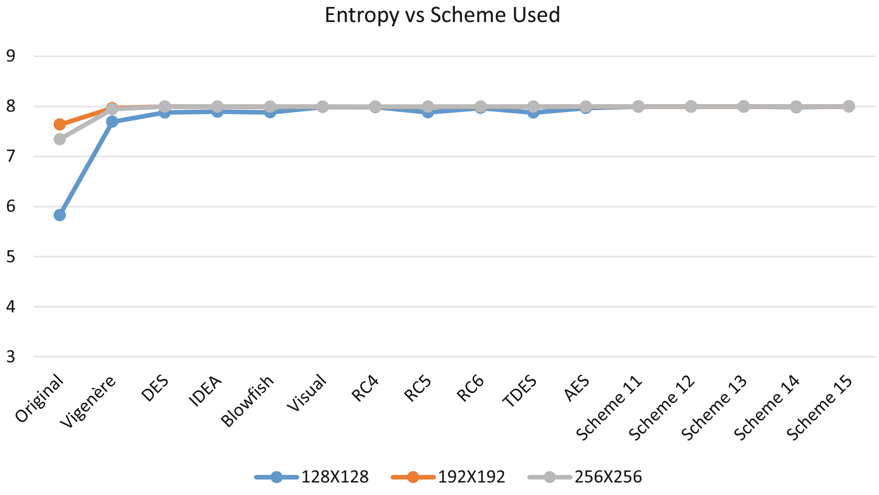


Fig. 21 Entropy values for plain and encrypted images



the scheme. Finally it should be observed that, unlike pixel sensitivity, the key sensitivity analysis depends on the key generation and has no dependency on the input image, so the NPCR and UACI values will not vary drastically with changes in the plain-image.

4.6 Quantitative Analysis

1. Peak signal to noise ratio (PSNR) analysis

Figure 20 shows the PSNR values obtained for the three test images. All the schemes used have almost similar values of PSNR for same test image. The PSNR values for the 128 × 128 test image are the highest, hence representing a comparatively easier

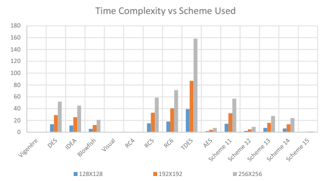


Fig. 22 Time complexities of the schemes used

data extraction for an unauthorized user as shown by other performance parameters.

2. Information entropy analysis

Figure 21 presents the entropy values obtained for the plain and encrypted images. It can be seen that for all three test images, almost all the schemes provide

Table 8 Summary of schemes used in the study according to results obtained by utilized performance metrics

Performance metrics/ encryption scheme	Visual scrambling	Histogram distribution	Correlation coefficient values	Pixel sensitivity	Key space	Key sensitivity	PSNR	Entropy	Time complexity
Vigenère	Least	Spiked	High	Least	Moderate	Low	Least	High	Least
DES	Moderate	Spiked	Moderate	Moderate	Low	High	High	High	High
IDEA	Moderate	Spiked	Moderate	Moderate	Moderate	High	High	High	High
Blowfish	Moderate	Spiked	Moderate	Moderate	Low	High	High	High	Moderate
Visual	High	Uniform	Very low	Least	Moderate	Least	High	High	Least
RC4	High	Uniform	Very low	Least	Moderate	High	High	High	Least
RC5	Moderate	Spiked	Moderate	Moderate	Moderate	High	High	High	High
RC6	Moderate	Spiked	Moderate	Moderate	Moderate	High	High	High	High
TDES	Moderate	Spiked	Moderate	Moderate	Moderate	High	High	High	Maximum
AES	Moderate	Spiked	Moderate	Moderate	Moderate	High	High	High	Low
Scheme 11	Very high	Uniform	Very low	Very high	Moderate	High	High	High	High
Scheme 12	Very high	Uniform	Very low	Very high	Moderate	High	High	High	Low
Scheme 13	Very high	Uniform	Very low	Very high	Moderate	High	High	High	Moderate
Scheme 14	Very high	Uniform	Very low	Very high	Maximum	High	High	High	Moderate
Scheme 15	Very high	Uniform	Very low	Very high	Maximum	High	High	High	Least

entropy values very close to the ideal value of 8. This value represents the resistance of the algorithms against entropy attack. Though the value was quite lesser when the 128 × 128 test image was encrypted by Vigenère scheme, the value was significantly higher than the entropy of plain-test-image.

4.7 Time Complexity Analysis

Figure 22 presents the time complexity for the schemes used in the study. The figure shows that some schemes like TDES, RC6 and scheme 11 have comparatively higher time complexities, hence loses an edge in applications where processing power is limited like for the mobile phone processor as compared to a personal computer or distributed computing processor.

5 Conclusion

The current study surveys ten conventional and five chaos-based encryption techniques to encrypt three test images of different size based on various performance metrics. Table 8 summarizes the schemes used in the study according to the results obtained by the metrics. It summarizes that the chaotic encryption schemes provides very high visually scrambled encrypted images with uniform histograms. Also these schemes provide very less correlation coefficient values in all the three directions. These parameters indicate their high resistance against the statistical attacks.

The chaotic schemes also have a high resistance against the differential attacks. This is because the schemes showed high pixel change and key change sensitivities. None of the conventional schemes was designed especially for images and hence none of them have any dependence on the initial image. So, these schemes showed poor pixel change sensitivities and hence will have low resistance against the differential attacks when these will be applied for image encryption.

All the schemes used in the study showed high information entropy values, hence ensuring no significant information leakage. Similarly, except Vigenère, all the encryption schemes showed similar PSNR values. Lastly, time complexity is one of the most essential criteria to assess performance of an encryption scheme. Conventional schemes like AES and RC4 along with chaos-based schemes like Scheme 15 provide very less time complexities and hence could be effective in cases where computational power or time is limited.

References

1. The next tier, 8 security predictions for 2017. https://www.trendmicro.ae/vinfo/ae/security/research-and-analysis/predictions/2017.
2. Bourbakis, N., & Alexopoulos, C. (1992). Picture data encryption using scan patterns. *Pattern Recognition, 25,* 567–581. https://doi.org/10.1016/0031-3203(92)90074-S.
3. Chang, C. C., Hwang, M. S., & Chen, T. S. (2001). A new encryption algorithm for image cryptosystems. *Journal of Systems and Software, 58,* 83–91. https://doi.org/10.1016/S0164-1212(01)00029-2.
4. Liu, Z., Guo, Q., Xu, L., Ahmad, M. A., & Liu, S. (2010). Double image encryption by using iterative random binary encoding in gyrator domains. *Optics Express, 18,* 12033–12043. https://doi.org/10.1364/OE.18.012033.
5. Tayal, N., Bansal, R., Gupta, S., & Dhall, S. (2016). Analysis of various cryptography techniques: A survey. *International Journal of Security and Its Applications, 10,* 59–92. https://doi.org/10.14257/ijsia.2016.10.8.07.
6. Zhang, G., & Liu, Q. (2011). A novel image encryption method based on total shuffling scheme. *Optics Communications, 284,* 2775–2780. https://doi.org/10.1016/j.optcom.2011.02.039.
7. Biham, E., & Shamir, A. (1993). *Differential cryptanalysis of the Data Encryption Standard*. Springer. http://www.cs.technion.ac.il/~biham/Reports/differential-cryptanalysis-of-the-data-encryption-standard-biham-shamir-authors-latex-version.pdf.
8. Zeghid, M., Machhout, M., Khriji, L., Baganne, A., & Tourki, R. (2007). A modified AES based algorithm for image encryption. *International Journal of Computer and Information Engineering, 1*(3), 745–750. http://scholar.waset.org/1307-6892/7580.
9. Chen, G., Mao, Y., & Chui, C. K. (2004). A symmetric image encryption based on 3D chaotic maps. *Chaos, Solitons & Fractals, 21,* 749–761. https://doi.org/10.1016/j.chaos.2003.12.022.
10. Dang, P. P., & Chau, P. M. (2000). Image encryption for secure internet multimedia applications. *IEEE Transactions on Consumer Electronics, 46,* 395–403. https://doi.org/10.1109/30.883383.
11. Younes, M. A. B., & Jantan, A. (2008). Image encryption using block-based transformation algorithm. *International Journal of Computer Science, 35*, 407–415. http://www.iaeng.org/IJCS/issues_v35/issue_1/IJCS_35_1_03.pdf.
12. Yun-peng, Z., Zheng-jun, Z., Wei, L., Xuan, N., Shui-ping, C., & Wei-di, D. (2009). Digital image encryption algorithm based on chaos and improved DES. In *IEEE international conference on systems, man, and cybernetics* (pp. 474–479). https://doi.org/10.1109/ICSMC.2009.5346839.
13. Akhshani, A., Akhavan, A., Lim, S. C., & Hassan, Z. (2012). An image encryption scheme based on quantum logistic map. *Communications in Nonlinear Science and Numerical Simulation, 17*(12), 4653–4661. https://arxiv.org/pdf/1307.7786.
14. García-Martínez, M., Denisenko, N., Soto, D., Arroyo, D., Orue, A., & Fernandez, V. (2013). High-speed free-space quantum key distribution system for urban daylight applications. *Applied Optics, 52*, 3311–3317. http://www.opticsinfobase.org/ao/upcomingpdf.cfm?id=185412.
15. Vidal, G., Baptista, M. S., & Mancini, H. (2012). Fundamentals of a classical chaos-based cryptosystem with some quantum cryptography features. *International Journal of Bifurcation and Chaos, 22*, Article number 1250243.
16. Kester, Q. A. (2013). A hybrid cryptosystem based on Vigenère cipher and columnar transposition cipher. *International Journal of Advanced Technology & Engineering Research, 3*, 141–147. https://arxiv.org/ftp/arxiv/papers/1307/1307.7786.pdf.
17. Matthews, R. (1989). On the derivation of a "chaotic" encryption algorithm. *Cryptologia, 14,* 29–42. https://doi.org/10.1080/0161-118991863745.
18. Mousa, A., & Hamad, A. (2006). Evaluation of the RC4 algorithm for data encryption. *International Journal of Computer Science & Applications, 3*, 44–56. http://www.tmrfindia.org/ijcsa/v3i24.pdf.
19. Matsui, M. (1994). The first experimental cryptanalysis of the Data Encryption Standard. *Advances in cryptology—CRYPTO'94* (pp. 1–11). https://doi.org/10.1007/3-540-48658-5_1.
20. Basu, S. (2011). International Data Encryption Algorithm (IDEA)—A typical illustration. *Journal of Global Research in Computer Science, 2*, 116–118. http://www.rroij.com/open-access/international-data-encryption-algorithm-idea-a-typical-illustration-116-118.pdf.
21. Schneier, B. (1994). The Blowfish encryption algorithm. *Dr. Dobb's Journal, 19*, 38–40. http://www.drdobbs.com/security/the-blowfish-encryption-algorithm/184409216.
22. Mandal, S., Das, S., & Nath, A. (2014). Data hiding and retrieval using visual cryptography. *International Journal of Innovative Research in Advanced Engineering, 1*, 102–110. http://ijirae.com/images/downloads/vol1issue2/ACS10107.April14.19.pdf.
23. Rivest, R. L. (1995). RC5 encryption algorithm. *Dr Dobb's Journal, 226*, 146–148. http://www.drdobbs.com/security/the-rc5-encryption-algorithm/184409480.
24. Li, C., Li, S., Asim, M., Nunez, J., Alvarez, G., & Chen, G. (2009). On the security defects of an image encryption scheme. *Image and Vision Computing, 27,* 1371–1381. https://doi.org/10.1016/j.imavis.2008.12.008.
25. Ahmed, H. E. H., Kalash, H. M., & Farag Allah, O. S. (2007). Encryption efficiency analysis and security evaluation of RC6 block cipher for digital images. In *International Conference on Electrical Engineering, 2007* (pp. 1–7). https://doi.org/10.1109/ICEE.2007.4287293.
26. Barker, W. C., & Barker, E. (2012). Recommendation for the Triple Data Encryption Algorithm (TDEA) block cipher. *National Institute of Standards and Technology, Special Publication* (pp. 800–867). https://doi.org/10.6028/NIST.SP.800-67r1.

27. Rayarikar, R., Upadhyay, S., & Pimpale, P. (2012). SMS encryption using AES algorithm on android. *International Journal of Computer Applications, 50*, 12–17. http://research.ijcaonline.org/volume50/number19/pxc3881038.pdf.
28. Sam, I. S., Devaraj, P., & Bhuvaneswaran, R. S. (2012). An intertwining chaotic maps based image encryption scheme. *Nonlinear Dynamics, 69,* 1995–2007. https://doi.org/10.1007/s11071-012-0402-6.
29. François, M., Grosges, T., Barchiesi, D., & Erra, R. (2012). A new image encryption scheme based on a chaotic function. *Signal Processing: Image Communication, 27,* 249–259. https://doi.org/10.1016/j.image.2011.11.003.
30. Sam, I. S., Devaraj, P., & Bhuvaneswaran, R. S. (2012). A novel image cipher based on mixed transformed logistic maps. *Multimedia Tools and Applications, 56,* 315–330. https://doi.org/10.1007/s11042-010-0652-6.
31. Hanchinamani, G., & Kulkarni, L. (2015). An efficient image encryption scheme based on a Peter De Jong chaotic map and a RC4 stream cipher. *3D Research*. https://doi.org/10.1007/s13319-015-0062-7.
32. Bansal, R., Gupta, S., & Sharma, G. (2016). An innovative image encryption scheme based on chaotic map and Vigenère scheme. *Multimedia Tools and Applications*. https://doi.org/10.1007/s11042-016-3926-9.
33. Anderson, T. W. (1958). *An introduction to multivariate statistical analysis*. New York: Wiley.
34. Anuradha, K., & Naik, P. P. S. (2015). Medical image cryptanalysis using histogram matching bitplane and adjoin mapping algorithms. *International Journal & Magazine of Engineering, Technology, Management and Research, 2*, 100–105. http://www.ijmetmr.com/oloctober2015/KolakaluriAnuradha-PPeddaSadhuNaik-13.pdf.
35. Karuvandan, V., Chellamuthu, S., & Periyasamy, S. (2016). Cryptanalysis of AES-128 and AES-256 block ciphers using Lorenz information measure. *The International Arab Journal of Information Technology, 13*, 306–312. http://ccis2k.org/iajit/PDF/Vol.13,%20No.3/5373.pdf.
36. Pareek, N. K., Patidar, V., & Sud, K. K. (2006). Image encryption using chaotic logistic map. *Image and Vision Computing, 24,* 926–934. https://doi.org/10.1016/j.imavis.2006.02.021.
37. Li, S., Zhao, Y., Qu, B., & Wang, J. (2012). Image scrambling based on chaotic sequences and Veginère cipher. *Multimedia Tools and Applications, 66,* 573–588. https://doi.org/10.1007/s11042-012-1281-z.
38. Xu, S., Wang, Y., & Wang, J. (2008). Cryptanalysis of two chaotic image encryption schemes based on permutation and XOR operations. In *International conference on computational intelligence and security* (pp. 433–437). https://doi.org/10.1109/CIS.2008.146.