**Sustainable Engineering and Innovation**
Vol. 3, No. 2, July 2021, pp.79-91
https://doi.org/10.37868/sei.v3i2.131

ISSN 2712-0562
*Original Research*

# Design and implementation of Threefish cipher algorithm in PNG file

**Ahmed S. Nori[1], Ansam O. Abdulmajeed[2]***
[1,2] College of Computer Science and Mathematics, University of Mosul, Mosul, Iraq

*Corresponding author: mbuco@student.ius.edu.ba



**Abstract**

This paper is presenting the design and implementation of the Threefish block cipher on grayscale images. Despite the fact that the Threefish block cipher is one of the best secure algorithms, most studies concerning Threefish have focused on hardware implementation and have not commonly been applied to image encryption due to the huge amount of data. The main contribution here was to reduce the time and the amount of data to be encrypted while maintaining encryption performance. This objective was achieved by encrypting just the most significant bits of image pixels. A 256-bit plain text block of the Threefish was constructed from 2n most significant bits of the pixels, where 0<n<3. Furthermore, a Threefish block cipher was applied when n=3 to analyze the impact of uninvolving some bits in the encryption process on the encryption performance. The results indicated that the encryption achieved good encryption quality when n=1, but it might cause some loss in decryption. In contrast, the encryption achieved high encryption quality when n=2, almost as good as the encryption of the whole pixel bits. Furthermore, the encryption time and the amount of data to be encrypted were decreased 50% as n decreased by 1. It was concluded that encrypting half of the pixel bits reduces both time and data, as well as significantly preserves the encryption quality. Finally, although the proposed method passed the statistical analysis, further work is needed to find a method resistant to the differential analysis for both colored and grayscale images.

*Keywords*: Threefish block cipher; Image encryption; Statistical analysis; Differential analysis

## 1. Introduction

Sensitive data is required to be transmitted in an inexplicable form by the intruders [1]. Cryptographic system gives a significant role in providing data security and maintain privacy. It is a system that mainly consists of four elements: encryption function, decryption functions, protocol, and key. These four elements determine the category and the strength of cryptographic system [2]. Based on the number of keys, cryptographic system can be classified into symmetric and asymmetric cryptographic system. In symmetric cryptography, the same secret key is used for both encryption and decryption. In contrast, asymmetric cryptography uses two different keys: one is public used for encryption, while the other is private for the recipient used for decryption [3]. Cryptographic system, in addition, can be classified into stream cipher and block cipher according to the input type of the data that will be encrypted. Stream cipher encrypts data bit/byte by bit/byte, whereas block cipher encrypts blocks of bits/bytes [1]. The DES (Data Encryption Standard), AES (Advance encryption Standard), Blowfish, Twofish, and Threefish, are symmetric block cipher algorithms, while RC5 (Rivest Cipher) is symmetric stream cipher algorithm. In addition, both RSA (Rivest Shamir Adleman) and ECC (Elliptic Curve

Cryptography) are asymmetric, where RSA is block cipher while ECC is stream cipher algorithm [3].

More than one study has examined the performance of different block cipher algorithms based on various parameters [3] [4]. These studies indicated that using large block size and extensive number of rounds made the Threefish algorithm one of the best secure algorithms. Threefish is considered tweakable block cipher, which is generalized form of block cipher. In tweakable block cipher, further input is used beside the input block and the key, which is known as tweak. Thereupon, the security level is increased [3] [4].

Threefish, however, is still not commonly applied on image encryption. Most of the studies used Threefish block cipher focused on hardware implementation on FPGA [5] [6] [7] [8]. Singh and Baburaj in 2018 proposed new image encryption method by combining Threefish algorithm and Artificial Neural Network in order to achieve high security and decrease the cost of computation [9]. As the image holds huge amount of data, our contribution here was to encrypt just the most significant bits of image pixels using Threefish block cipher in order to reduce the time and the amount of data to be encrypted while maintaining encryption quality. In this way, the proposed method encrypts half data of the image or less, and preserves the encryption performance at the same time.

## 2. Materials and method

### 2.1. Threefish

Threefish uses three different key lengths: 256, 512, or 1024 bits. In this algorithm, block size is identical to the key length used [4]. Threefish algorithm uses 128 bits tweak value regardless to the block size and key length. Threefish is adopted to use modulus arithmetic, bit rotation, and bitwise XOR. These operations are applied several rounds depending on the block size. Block size of 256 and 512 bits consist of 72 rounds, while block size of 1024 bits consists of 80 rounds [5]. In each round, Threefish operates on 64-bit unsigned integers, that is the plain text is divided into $N_w$ words of 64-bit where [6]:

$$N_w = Block\ size/64 \tag{1}$$

#### 2.1.1. Threefish key scheduling

Threefish generates $N_r$ /4+1 subkeys from the cipher key, where $N_r$ is the number of rounds. Along with the cipher key $K$, Threefish uses the 128-bit tweak value $T$ and 64-bit constant value $C_{240}$ to produce these subkeys ($K_0, K_1, \ldots, K_{Nw-1}$). Prior to start the subkeys scheduling, the two 64-bit words of tweak value ($t_0, t_1$) are extended to further word $t_2$. In addition, the 64-bit words of the original key ($K_0, K_1, \ldots, K_{Nw-1}$) are used to extend the $K_{Nw}$ key word as the following:

$$t_2 = t_0 \oplus t_1 \tag{2}$$

$$K_{N_w} = C_{240} \oplus K_0 \oplus \ldots \oplus K_{N_{w-1}} \tag{3}$$

The subkeys in every round are defined as the following:

$$K_{s,i} = \begin{cases} K_{(s+i)} mod\ (N_w + 1) & i = 0, \ldots, N_w - 4 \\ K_{(s+i)} mod\ (N_w + 1) \boxplus t_{s\ mod\ 3} & i = N_w - 3 \\ K_{(S+1)} mod\ (N_w + 1) \boxplus t_{s\ mod\ 3} & i = N_w - 2 \\ K_{(s+1)} mod\ (N_w + 1) \boxplus s & i = N_w - 1 \end{cases} \tag{4}$$

where $0 \le i \le N_w$ -1, and $0 \le s \le N_r / 4$, the symbol $\oplus$ denotes bitwise xor operation, and $\boxplus$ denotes to addition modulo $2^{64}$ [3], [4], [7].

### 2.1.2. Threefish encryption

Encryption in Threefish block cipher starts by adding the subkey to the plain words. This operation is repeated every four rounds. Every round consists of Mix operations and permutation. Fig. 1 shows four rounds of Threefish256 block cipher [7].

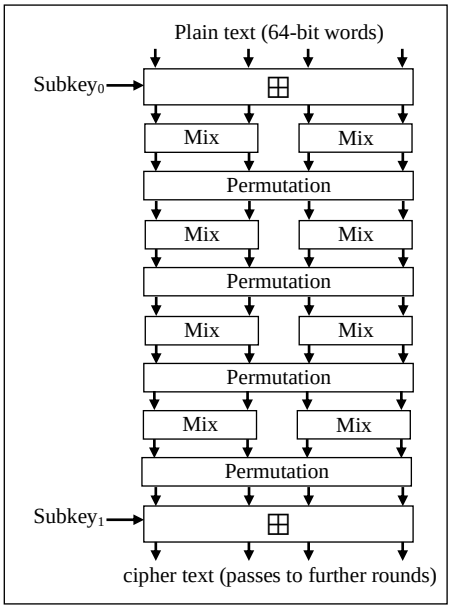



Figure 1. Four rounds of Threefish256 encryption [7]

Every Mix operation operates on two 64-bit words as the following: (see Fig. 2):

$$f_{s,0} = e_{s,0} \boxplus e_{s,1} \quad (5)$$

$$f_{s,1} = (e_{s,1} \lll R_{d,j}) \oplus f_{s,0} \quad (6)$$

where d = s mod 8, and the expression $\lll R$ indicates to bit rotate left R times, where R is a constant value as listed in Table 1 [5].

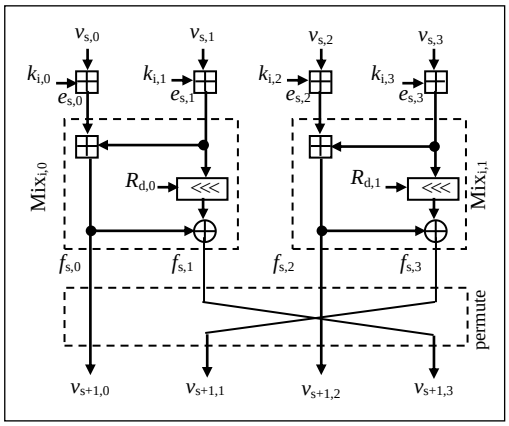



Figure 2. Single round of Threefish256 encryption [8]

Table 1. R values in Threefish 256 [5]

| j \ d | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 0 | 14 | 52 | 23 | 5 | 25 | 46 | 58 | 32 |
| 1 | 16 | 57 | 40 | 37 | 33 | 12 | 22 | 32 |

The 64-bits words resulted from Mix operations is permutated, as listed in the Table 2, to produce the 64-bits cipher words that pass to the next round. Prior to the end of encryption process, Threefish256 block cipher performs a subkey addition to produce the final cipher words after 72 rounds [8].

Table 2. Words permutation in Threefish 256 [8]

| | | | | |
|---|---|---|---|---|
| Word number | 0 | 1 | 2 | 3 |
| Word number after permutation | 0 | 3 | 2 | 1 |

### 2.1.3. Threefish decryption

In Threefish, decryption includes the same steps of encryption, but in reverse order. Fig. 3 shows single round of Threefish256 decryption.

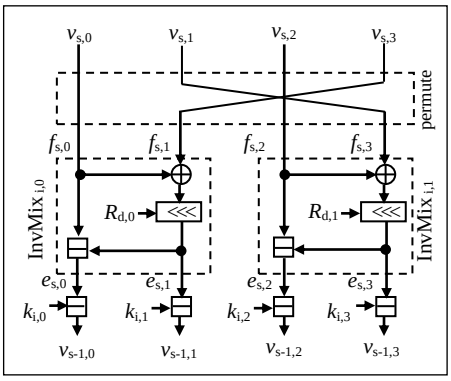



Figure 3. Single round of Threefish256 decryption [8]

Threefish256 starts by subtracting the subkeys from the cipher text words. Then, the 72 rounds started by permutating the cipher words according to Table 2. These permutated words passed to the inverse Mix operation as the following:

$$e_{s,1} = \left(f_{s,0} \oplus f_{s,1}\right) \ggg R_{d,0} \tag{7}$$

$$e_{s,0} = f_{s,0} \boxminus e_{s,1} \tag{8}$$

where $\ggg R_{d,0}$ indicates to bit rotate right R times according to Table 1, and $\boxminus$ is subtraction modulo $2^{64}$. In every four rounds, the subkey is subtract from the cipher words in reverse order. These operations are repeated until getting the plain words after 72 rounds [7].

### 2.2. The proposed method

According to the concepts of Threefish256, the proposed method encrypts PNG grayscale images. The encryption is limited to the most $2^n$ significant bits of pixel image because those bits has significantly affected the image quality. The proposed method adopted n=1 and n=2. For grayscale image pixels "156 159 158 159…...", the plain text block that generated when n=1 is "10101010…...", while it is "1001100110011001…..." when n=2.

The 256-bits plain blocks were generated according to the following steps:

1. Determining the value of n.
2. Getting the $2^n$ most significant bits from each pixel.
3. Collecting those bits in a matrix, let's naming it WORDS, of 256 × X where:

$$X = (imageRow \times imageCol \times 2^n) \div 256 \tag{9}$$

4. Converting each 64 bits of the WORDS matrix into unsigned integer 64 number.

The constant value of $C_{240}$ used key scheduling was 1BD11BDAA9FC1A22 in hexadecimal. The proposed method used 32-characters phrase as a secret key and 16-characters phrase as a tweak value. "Fig. 4" shows the dataflow of Threefish256 encryption of the proposed method.

The decryption follows the same steps but with flipped subkeys. The 256-bits cipher blocks were made up the WORDS matrix according to the previously mentioned steps. Then, these 64-bit blocks were entered to the decryption process along with the subkeys that were scheduled using the same key and tweak value that is used in encryption. In decryption, the subkeys were used in reverse order, i.e., in round 1 the subkey 18 was subtracted from the cipher blocks.

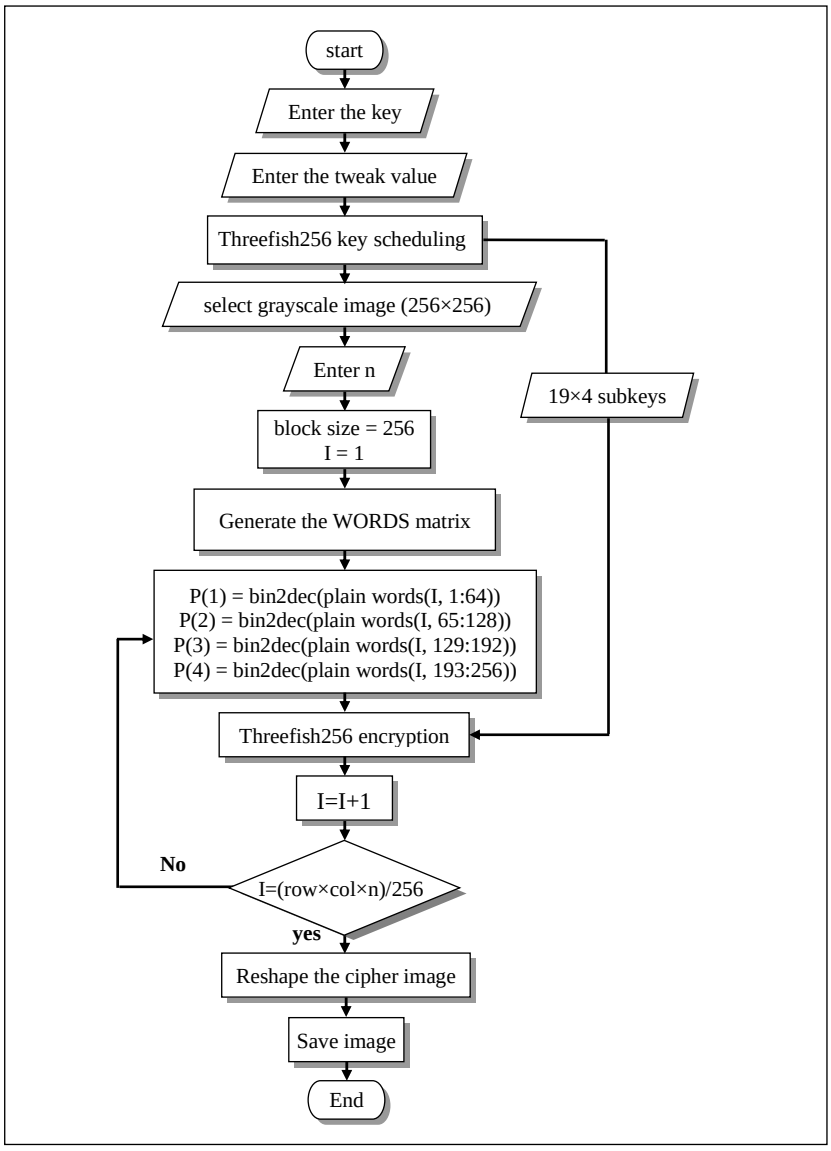



Figure 4. The dataflow of proposed method

The image cryptographic system is resistant to statistical analysis if it is infeasible to predict the key or the plain image according to the distribution of grayscales in cipher image [10]. The most common statistical analysis tests are histogram variance, entropy, contrast, and energy. The cipher image histogram should be uniform as much as possible. The histogram uniformity was measured by variance, which is calculated as the following:

$$v(n) = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \left(h_i - h_j\right)^2 \quad (10)$$

where $h_i$ is the value of cipher image histogram of pixel value $i$ [11]. the lesser the variance, the more uniform the histogram [12]. The entropy estimates the randomness amount in the cipher image. It is calculated as the following [13]:

$$H(s) = - \sum_{i=0}^{2^n-1} p(s_i) \log_2[p(s_i)] \quad (11)$$

where $p(s_i)$ is the probability of value $s_i$. The higher the entropy, the more randomly pixels distribution is. For 8-bit grayscale image, the ideal value of the entropy is 8 [12].

The contrast refers to the difference in intensity among adjacent pixels in image. A good encrypted image should have high contrast. The contrast is calculated as the following:

$$Contrast = \sum |i-j|^2 \, p(i,j) \quad (12)$$

where *p*(*i*; *j*) is pixel position in GLCM (Gray Level Co-occurrence Matrix ) matrix [1].The energy evaluates the change rate in pixel brightness. The energy can be computed as the following:

$$Energy = \sum \eta(i,j)^2 \tag{13}$$

where, ŋ (i; j) is the number of GLCM matrices. The lowest the energy the more secure cipher image is [1]. PSNR (Peak Signal-to-Noise Ratio) estimates the distortion between plain and cipher image. the lower the value of PSNR, the more secure encryption is. PSNR can be computed as the following:

$$PSNR = 20 \times \log_{10}\left[\frac{255}{\sqrt{MSE}}\right] \tag{14}$$

where MSE, the Mean Square Error, can be computed as the following:

$$MSE = \frac{1}{M \times N} \sum_{n=1}^{N} \sum_{m=1}^{M} (p(m,n) - c(m,n))^2 \tag{15}$$

where *p* and *c* are the plain and cipher image, respectively. The higher the MSE, the more secure encryption is [1], [11]. Cipher image should not be correlated with the neighbored pixels whether in diagonal, vertical, or horizontal directions. For the two vectors *u* and *v*, the correlation coefficients can be computed as the following [11]:

$$r_{xy} = \frac{cov(u,v)}{\sqrt{D(u)}\sqrt{D(v)}} \tag{16}$$

$$cov(u,v) = \frac{1}{N} \sum_{i=1}^{N} \big(x_i - E(u)\big)\big(y_i - E(v)\big) \tag{17}$$

$$D(u) = \frac{1}{N} \sum_{i=1}^{N} (u_i - E(u))^2 \tag{18}$$

$$E(u) = \frac{1}{N} \sum_{i=1}^{N} u_i \tag{19}$$

In the plain image, the correlation coefficients are near to 1, while they are close to 0 in cipher image [13].
The image cryptographic system is resistant to differential analysis if any tiny changes in the plain image can, significantly, affects the cipher image [11]. NPCR (Number of Pixels Change Rate) is used to compute the change rate in pixel values at specific position of two cipher images when a single value differs in the corresponding plain image. NPCR is calculated as the following:

$$NPCR = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} |Sign\,(C_1(i,j) - C_2(i,j))|}{MN} \tag{20}$$

where,

$$Sign(x) = \begin{cases} 1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases} \tag{21}$$

$C_1$ is the cipher image, and $C_2$ is the cipher image of the same plain image, but with one-pixel value differs [11]. UACI (Unified Average Changing Intensity) is used to measure the average of the difference of the pixels in specific position to the maximum difference. It can be computed as the following [11]:

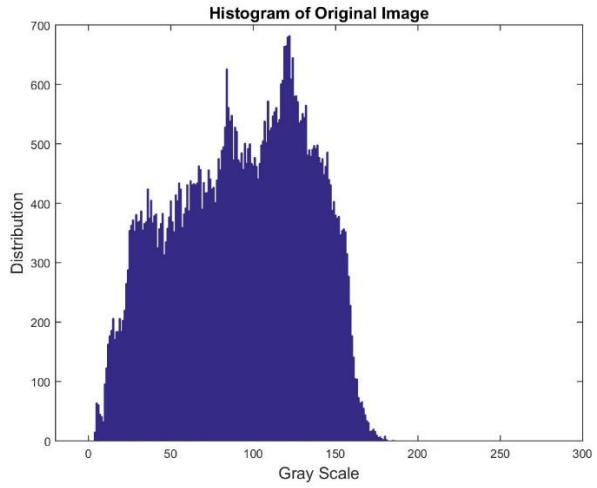
$$UACI = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{|C_1(i,j) - C_2(i,j)|}{255-0} \tag{22}$$

The ideal value of each NPCR and UACI are 99.6094% and 33.4635% respectively [12].

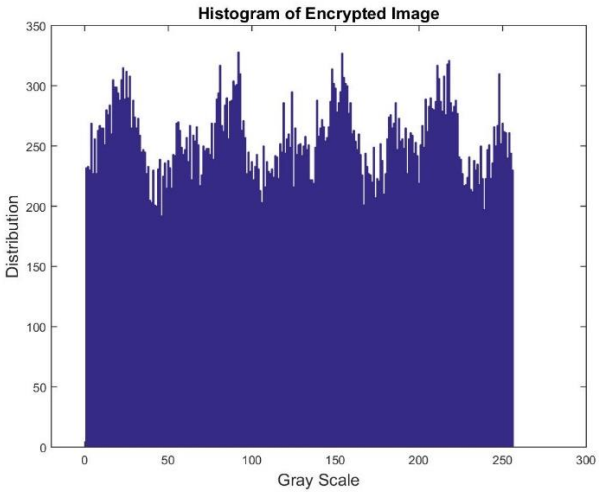
## 3. Results and discussion

More than 10 PNG grayscale images were used to analyze the security of the proposed method using MATLAB. The experiments included applying Threefish256 block cipher in the case of n=1 and n=2.

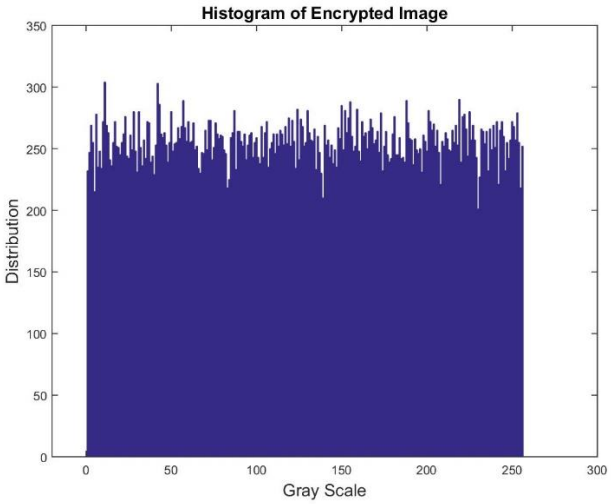
Furthermore, Threefish256 block cipher was applied when n=3 to analyze the impact of uninvolving some bits in encryption process on the encryption performance. In the statistical analysis, histogram variance was used to measure the uniformity of histogram, entropy was used to measure the randomness, contrast, and energy were used to measure the differences in intensity and brightness of the cipher image. Fig. 5 shows the histogram of the plain and cipher images for one of the tested images.



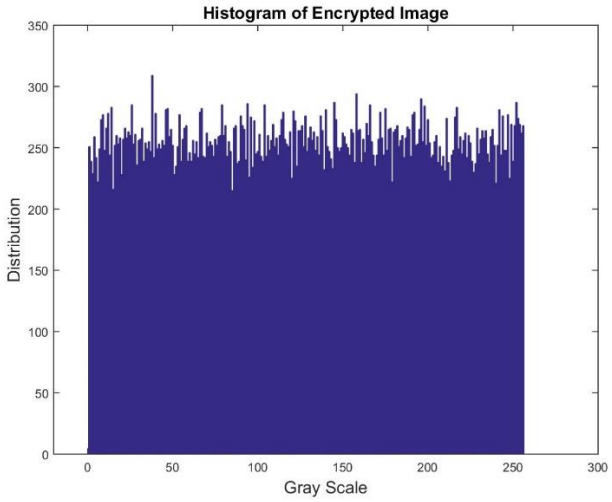

Plain image histogram




Cipher image histogram, n=1




Cipher image histogram, n=2




Cipher image histogram, n=3

Figure 5. "Zelda.png" plain and cipher images histogram

The histogram uniformity measured according to "(10),". Table 3 shows the histograms variance for both plain and cipher images in the three cases of *n*. The variance of the cipher images in the case of n=2 is high and close to those in the case of n=3, while it is high acceptant when n=1 for most of the tested images. The entropy value of the cipher images of the proposed method for the three cases of *n* was close to the ideal value (see Table 3).

Table 3 shows that the cipher images of the proposed method have high contrast and low energy compared to their corresponding plain images. The contrast and energy in the case of n=2 was also close to those in case n=3.

Table 3. The variance, entropy, contrast and energy analysis of the proposed method

| Image name | n | Variance (plain) | Variance (cipher) | Entropy (plain) | Entropy (cipher) | Contrast (plain) | Contrast (cipher) | Energy (plain) | Energy (cipher) |
|---|---|---|---|---|---|---|---|---|---|
| Barbara | 1 | 41188.3516 | 1978.9141 | 7.4184 | 7.9788 | 0.3443 | 9.8482 | 0.1091 | 0.019733 |
| | 2 | | 358.9766 | | 7.9960 | | 10.2668 | | 0.015663 |
| | 3 | | 254.2422 | | 7.9972 | | 10.4226 | | 0.015641 |
| Cameraman | 1 | 110973.3047 | 10001.6172 | 7.0097 | 7.8954 | 0.5872 | 7.76651 | 0.1805 | 0.027983 |

| Image name | n | Variance (plain) | Variance (cipher) | Entropy (plain) | Entropy (cipher) | Contrast (plain) | Contrast (cipher) | Energy (plain) | Energy (cipher) |
|---|---|---|---|---|---|---|---|---|---|
| | 2 | | 878.25 | | 7.9904 | | 9.9709 | | 0.015904 |
| | 3 | | 253.6406 | | 7.9972 | | 10.4777 | | 0.01564 |
| | 1 | | 51953.8984 | | 7.7069 | | 9.7574 | | 0.025487 |
| Clown | 2 | 222998.8906 | 12029.66 | 7.15859 | 7.9028 | 0.4186 | 10.3301 | 0.1817 | 0.015669 |
| | 3 | | 304.9766 | | 7.9966 | | 10.3712 | | 0.015652 |
| | 1 | | 14848.0781 | | 7.8369 | | 8.9859 | | 0.031254 |
| Girl face | 2 | 71484.2578 | 1568.672 | 7.2541 | 7.9830 | 0.2019 | 10.2985 | 0.1621 | 0.015657 |
| | 3 | | 228.5781 | | 7.9975 | | 10.4993 | | 0.015637 |
| | 1 | | 648.6719 | | 7.9929 | | 9.1812 | | 0.022322 |
| Lena | 2 | 41143.9297 | 389.2422 | 7.4318 | 7.9957 | 0.3353 | 10.4592 | 0.1236 | 0.015654 |
| | 3 | | 233.3594 | | 7.9974 | | 10.4688 | | 0.015638 |
| | 1 | | 5883.3828 | | 7.9425 | | 10.2582 | | 0.020582 |
| Man | 2 | 37038.6484 | 766.6641 | 7.536 | 7.9916 | 0.4743 | 10.4991 | 0.1005 | 0.015642 |
| | 3 | | 262.7813 | | 7.9971 | | 10.4282 | | 0.01564 |
| | 1 | | 2043.203 | | 7.9782 | | 10.3866 | | 0.017028 |
| Mandrill | 2 | 53931.1016 | 305.0938 | 7.2937 | 7.9967 | 0.6762 | 10.4577 | 0.0902 | 0.015645 |
| | 3 | | 283.9766 | | 7.9969 | | 10.5381 | | 0.015643 |
| | 1 | | 2515.938 | | 7.9725 | | 10.0008 | | 0.022537 |
| Pepper | 2 | 31580.2109 | 302.0781 | 7.5807 | 7.9967 | 0.2975 | 10.4813 | 0.1134 | 0.015639 |
| | 3 | | 250.0781 | | 7.9973 | | 10.5151 | | 0.015635 |
| | 1 | | 817.0391 | | 7.9911 | | 9.8289 | | 0.02283 |
| Zelad | 2 | 50175.6406 | 247.4063 | 7.2523 | 7.9973 | 0.1897 | 10.3832 | 0.1492 | 0.015644 |
| | 3 | | 243.8672 | | 7.9973 | | 10.4281 | | 0.015637 |

The result of PSNR and MSE are listed in Table 4, where the cipher images had low PSNR and high MSE in all cases of *n*, that means that cipher images are highly different from the corresponding plain images. Fig. 6 shows the encrypted images in the cases n=1, n=2, and n=3.

Table 4. The PSNR and MSE analysis of the proposed method

| Image name | n | PSNR | MSE | MSE (Decipher) |
|---|---|---|---|---|
| | 1 | 8.5906 | 7835.9375 | 0 |
| Barbara | 2 | 8.639 | 7749.094 | 0 |
| | 3 | 8.6704 | 7693.396 | 0 |
| | 1 | 8.8528 | 8336.0625 | 36 |
| Cameraman | 2 | 8.2877 | 9494.516 | 0 |
| | 3 | 8.3246 | 9414.228 | 0 |
| | 1 | 7.4514 | 11602 | 342 |
| Clown | 2 | 6.9664 | 12972.48 | 0 |
| | 3 | 6.8891 | 13205.65 | 0 |
| | 1 | 7.0504 | 8780.5625 | 0 |
| Girl face | 2 | 6.5519 | 9848.586 | 0 |
| | 3 | 6.4730 | 10029.06 | 0 |
| | 1 | 8.4709 | 8191 | 414 |
| Lena | 2 | 8.7415 | 7696.168 | 0 |
| | 3 | 8.7112 | 7750.1 | 0 |
| | 1 | 7.6460 | 9335 | 0 |
| Man | 2 | 7.3337 | 10030.95 | 0 |
| | 3 | 7.2899 | 10132.59 | 0 |
| | 1 | 8.8843 | 6958.938 | 0 |
| Mandrill | 2 | 8.8485 | 7016.605 | 0 |
| | 3 | 8.8395 | 7031.075 | 0 |
| | 1 | 8.0645 | 7975.75 | 0 |
| Pepper | 2 | 7.8655 | 8349.648 | 0 |

| Image name | n | PSNR | MSE | MSE (Decipher) |
|---|---|---|---|---|
| | 3 | 7.8460 | 8387.217 | 0 |
| Zelad | 1 | 6.0243 | 8456.813 | 0 |
| | 2 | 6.0697 | 8368.805 | 0 |
| | 3 | 6.0430 | 8420.362 | 0 |

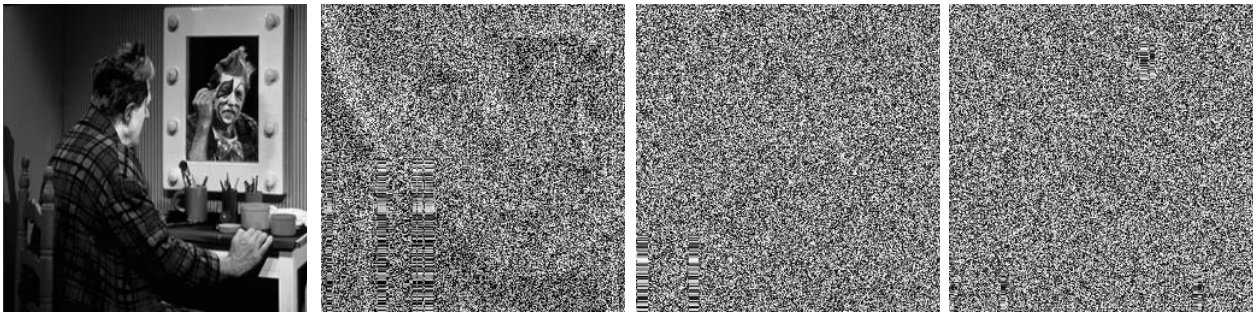



Figure 6. "Clown.png" cipher images according to the proposed method

The nonzero MSE for the decipher images indicates that there is a loss in some decipher images in case of n=1. Fig. 7 shows the cipher and decipher images in the three cases of n.

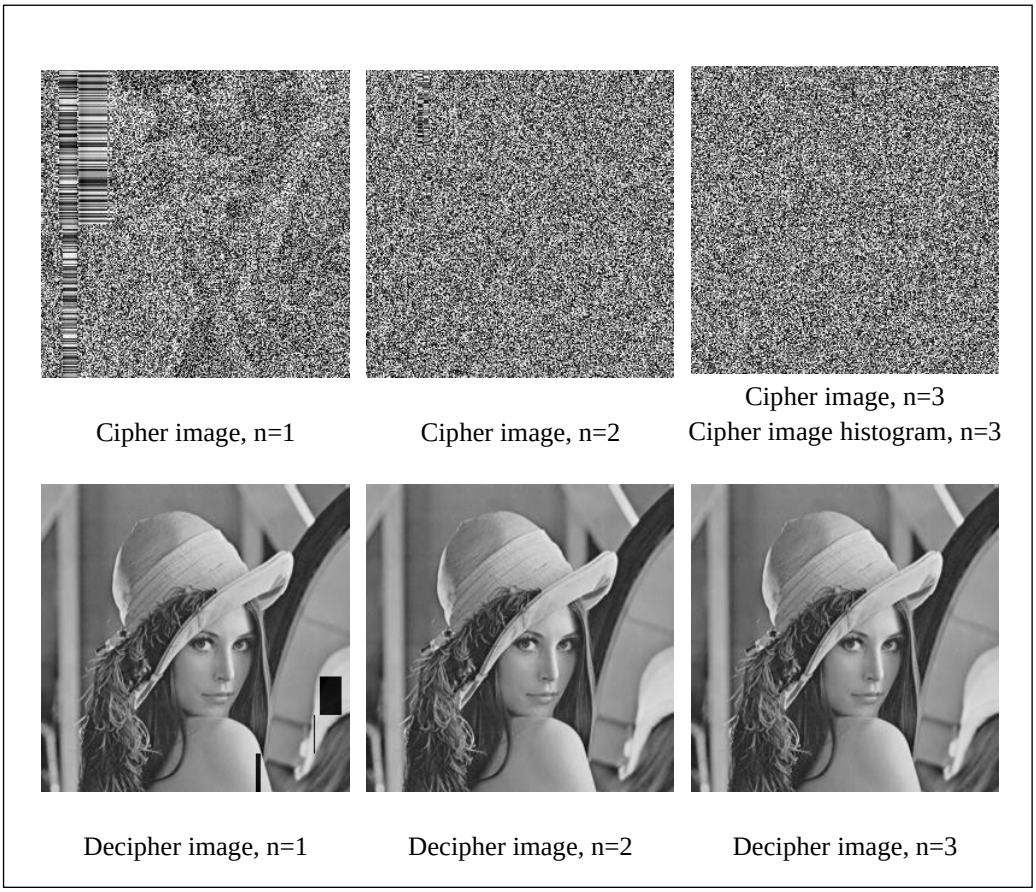


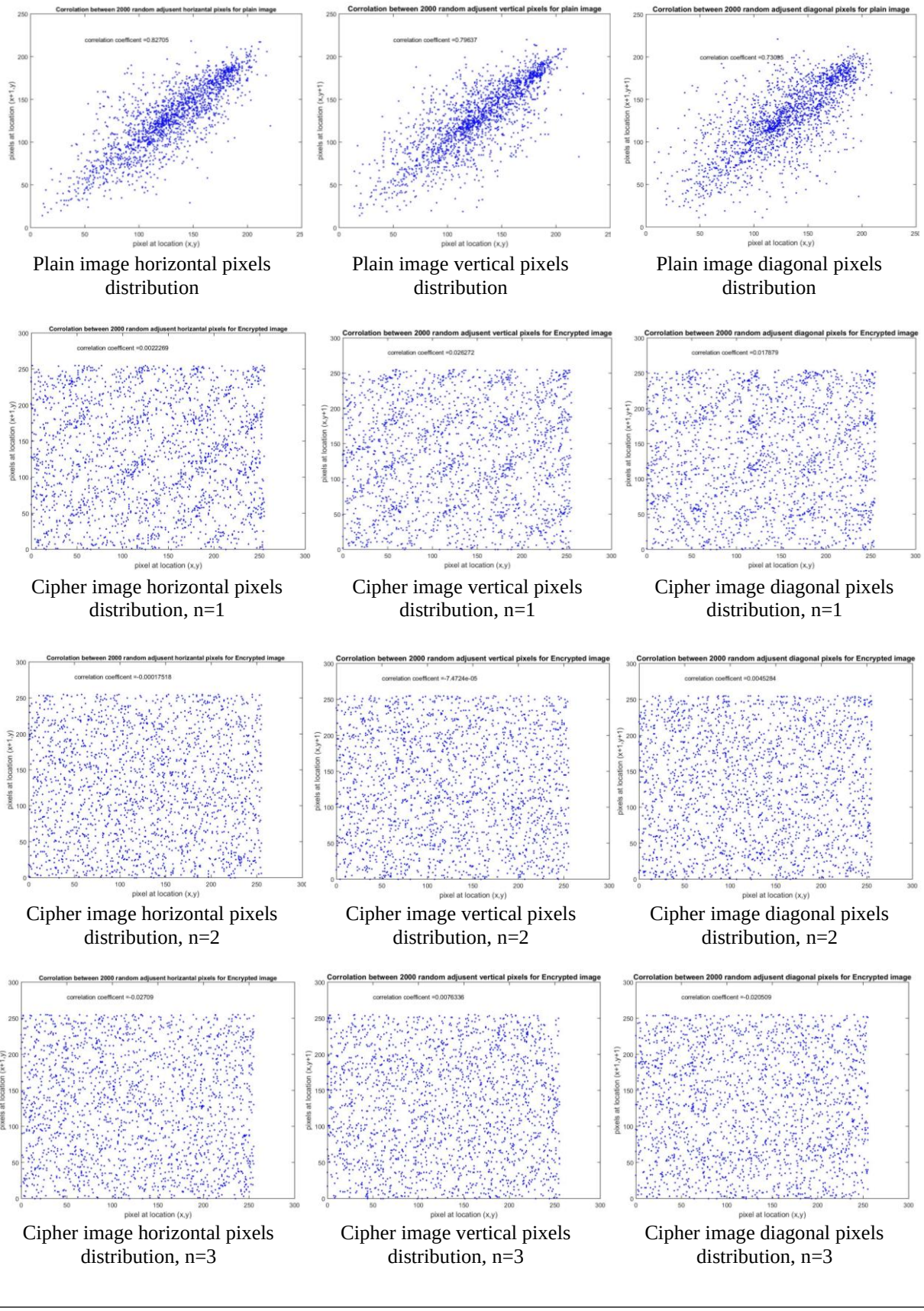

Figure 7. "Lena.png" cipher and decipher images in all the case of n

The correlation coefficients of 2000 random pixels on horizontal, vertical, and diagonal are listed in Table 5. The correlation coefficients of the cipher images were close to 0, which means that the adjacent pixels are highly décorrelated to each other.

Table 5. The correlation coefficients of the plain and cipher images in all the cases of N

| Image name | N | H (Plain) | H (Cipher) | V (Plain) | V (Cipher) | D (Plain) | D (Cipher) |
|---|---|---|---|---|---|---|---|
| Barbara | 1 | 0.9402 | 0.0691 | 0.9651 | 0.0318 | 0.9188 | 0.0040 |
| | 2 | 0.9400 | 0.0350 | 0.9644 | 0.0024 | 0.9230 | -0.0033 |
| | 3 | 0.9470 | 0.0225 | 0.9642 | -0.0353 | 0.9191 | -0.0142 |
| Cameraman | 1 | 0.9319 | 0.2481 | 0.9584 | 0.0190 | 0.9221 | 0.0936 |
| | 2 | 0.9327 | 0.0273 | 0.9578 | -0.0201 | 0.9055 | -0.0365 |
| | 3 | 0.9435 | -0.0106 | 0.9542 | 0.0051 | 0.9145 | 0.0262 |
| Clown | 1 | 0.9559 | 0.0728 | 0.9747 | 0.0155 | 0.9362 | 0.0036 |
| | 2 | 0.9569 | -0.0001 | 0.9790 | 0.0212 | 0.9365 | 0.0292 |
| | 3 | 0.9583 | 0.0438 | 0.9788 | -0.0461 | 0.9354 | -0.0191 |
| Girl face | 1 | 0.9715 | 0.1471 | 0.9768 | 0.0142 | 0.9545 | 0.0685 |
| | 2 | 0.9718 | -0.0140 | 0.9740 | -0.0039 | 0.9428 | -0.0238 |
| | 3 | 0.9691 | 0.0013 | 0.9789 | 0.0105 | 0.9469 | 0.0013 |
| Lena | 1 | 0.9485 | 0.1507 | 0.9754 | 0.0547 | 0.9266 | 0.0260 |
| | 2 | 0.9429 | 0.0265 | 0.9735 | -0.0169 | 0.9121 | 0.0201 |
| | 3 | 0.9522 | 0.0052 | 0.9731 | -0.0068 | 0.9256 | 0.0129 |
| Man | 1 | 0.9437 | -0.0015 | 0.9566 | 0.0247 | 0.9260 | 0.0277 |
| | 2 | 0.9430 | -0.0157 | 0.9547 | 0.0076 | 0.9035 | 0.0069 |
| | 3 | 0.9376 | 0.0084 | 0.9570 | -0.0097 | 0.9106 | -0.0227 |
| Mandrill | 1 | 0.8271 | 0.0022 | 0.7963 | 0.0262 | 0.7309 | 0.0178 |
| | 2 | 0.8434 | -0.0002 | 0.8025 | -0.00007 | 0.7083 | 0.0045 |
| | 3 | 0.8311 | -0.0271 | 0.7688 | 0.0076 | 0.7448 | -0.0205 |
| Pepper | 1 | 0.9616 | 0.0605 | 0.9649 | 0.0608 | 0.9447 | 0.0229 |
| | 2 | 0.9646 | -0.0228 | 0.9698 | 0.0117 | 0.9377 | -0.0179 |
| | 3 | 0.9637 | 0.0058 | 0.9742 | -0.0149 | 0.9366 | -0.0194 |
| Zelad | 1 | 0.9748 | 0.0656 | 0.9839 | 0.0413 | 0.9592 | 0.0077 |
| | 2 | 0.9706 | -0.0069 | 0.9816 | 0.0060 | 0.9587 | -0.0157 |
| | 3 | 0.9739 | -0.0451 | 0.9832 | -0.0189 | 0.9589 | 0.0061 |

The distribution of the 2000 adjacent pixels of the plain and the cipher images of one of the tested images are shown in Fig. 8.




Plain image horizontal pixels distribution | Plain image vertical pixels distribution | Plain image diagonal pixels distribution

Cipher image horizontal pixels distribution, n=1 | Cipher image vertical pixels distribution, n=1 | Cipher image diagonal pixels distribution, n=1

Cipher image horizontal pixels distribution, n=2 | Cipher image vertical pixels distribution, n=2 | Cipher image diagonal pixels distribution, n=2

Cipher image horizontal pixels distribution, n=3 | Cipher image vertical pixels distribution, n=3 | Cipher image diagonal pixels distribution, n=3

Figure 8. The distribution of the adjacent pixels of the plain and the cipher images of "Mandrill.png"

Both of NPCR and UACI tests are computed between two cipher images of the same plain image but with one-pixel value difference. The values of NPCR and UACI, listed in Table 6, are far from the ideal values

because the change in a single pixel affected the corresponding 256-bit block but not the whole image. Furthermore, the value of NPCR and UACI decreased as the value of n increased; because not all the pixel bits are involved in the encryption when n < 3.

Table 6. NPCR and UACI analysis of the proposed method in all the case of N

| Image name | n | NPCR | UACI |
|---|---|---|---|
| Barbara | 1 | 0.1480 | 0.0620 |
| | 2 | 0.0916 | 0.0342 |
| | 3 | 0.0488 | 0.0201 |
| Cameraman | 1 | 0.1312 | 0.0597 |
| | 2 | 0.0916 | 0.0336 |
| | 3 | 0.0488 | 0.0166 |
| Clown | 1 | 0.1450 | 0.0548 |
| | 2 | 0.0931 | 0.0327 |
| | 3 | 0.0488 | 0.0153 |
| Girl face | 1 | 0.1602 | 0.0624 |
| | 2 | 0.0900 | 0.0338 |
| | 3 | 0.0488 | 0.0160 |
| Lena | 1 | 0.1434 | 0.0555 |
| | 2 | 0.0900 | 0.0271 |
| | 3 | 0.0488 | 0.0128 |
| Man | 1 | 0.1434 | 0.0567 |
| | 2 | 0.0916 | 0.0348 |
| | 3 | 0.0488 | 0.0181 |
| Mandrill | 1 | 0.1450 | 0.0586 |
| | 2 | 0.0946 | 0.0343 |
| | 3 | 0.0488 | 0.0178 |
| Pepper | 1 | 0.1511 | 0.0643 |
| | 2 | 0.0854 | 0.0310 |
| | 3 | 0.0473 | 0.0113 |
| Zelad | 1 | 0.1602 | 0.0666 |
| | 2 | 0.0961 | 0.0367 |
| | 3 | 0.0458 | 0.0180 |

The average encryption and decryption time of the proposed method in all cases of n are listed in Table 7, which indicated that the time is decreased by 50% as n decreased by 1. It is worth mentioned that the number of 256-bit blocks of 256×256 grayscale image when n=3 is 2048 blocks, while it is decreases to 1024 blocks when n=2, and 512 blocks when n=1.

Table 7. Encryption and decryption time of the proposed method

| N | Encryption Time (Sec.) | Decryption Time (Sec.) |
|---|---|---|
| 1 | 4.33754 | 4.121164 |
| 2 | 7.853798 | 7.502774 |
| 3 | 15.42821 | 14.56677 |

## 4. Conclusion

The objective of the current work was to design and implement Threefish block cipher on grayscale images by applying the encryption just on the $2^n$ most significant bits of image pixels to reduce the time and the amount of data to be encrypted while maintaining encryption performance. The results showed that the encryption of just the $2^1$ most significant bits achieves good encryption quality but it may cause some loss in decryption, while the encryption of the $2^2$ most significant bits achieves high encryption quality almost as good as the encryption of the total bits. Furthermore, the encryption time and the amount of data to be encrypted are decreased to 50% as n decreases. Encrypting the $2^2$ most significant bits instead of encrypting total bits is sufficient to preserve high encryption quality, as well as reduces the time and the data to be encrypted. The proposed method resists the statistical analysis; however, further work to find a method resistant to the differential analysis for both colored and grayscale images is recommended.

**References**


[1] A. H. Zahid, E. Al-Solami, and M. Ahmad, "A Novel Modular Approach Based Substitution-Box Design for Image Encryption," *IEEE Access*, vol. 8, pp. 150326-150340, 2020.

[2] S. Hussain, S. S. Jamal, T. Shan, and I. Hussain, " A Power Associative Loop Structure for the Construction of Non-Linear Components of Block Cipher," *IEEE Access*, vol. 8, pp. 123492-123506, 2020.

[3] R. Bhanot, and R. Hans "A Review and Comparative Analysis of Various Encryption Algorithms," *International Journal of Security and Its Applications*, vol. 9, no. 4, pp. 289–306, 2015.

[4] S. W. Jang, "Comparative Analysis of AES, Blowfish, Twofish and Threefish Encryption Algorithms," *Analysis of Applied Mathematics*, vol. 10, pp. 5-24, 2017.

[5] K. M. Mhaidat, M. A. Altahat, and O. D. Al-Khaleel, "High-Throughput Hardware Implementation of Threefish Block Cipher on FPGA," in *International Conference on Information and Communication Systems.*, Irbid, Jordan, 2013.

[6] N. At, J. Beuchat, and I. San, "Compact Implementation of Threefish and Skein on FPGA," in *5th IFIP International Conference on New Technologies, Mobility and Security*, IEEE Press, 2012.

[7] L. P. Oommen, and Anas A. S., "Skein and Threefish Implementation on FPGA," *International Journal of Science and Research (IJSR)*, vol. 4, no. 5, pp.1493-1496, 2015.

[8] P. Gayathri, K. Sateesh, and C. Navya, "High-Throughput Hardware Implementation of Three Fish Block Cipher Encryption and Decryption on FPGA," *International Journal of VLSI System Design and Communication Systems*, vol. 3, no. 8, pp. 1325-1329, 2015.

[9] J. J. Singh. T, and E. Baburaj, "A Novel Method for Secured Transaction of Images and Text on cloud," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 11, pp. 506-511, 2018.

[10] P. T. Akkasaligar, and S. Biradar, " Selective medical image encryption using DNA cryptography," *Information Security Journal: A Global Perspective*, vol. 29, no. 2, pp. 91-101, 2020.

[11] L. Liu, L. Zhang, D. Jiang, Y. Guan, and Z. Zhang, "A Simultaneous Scrambling and Diffusion Color Image Encryption Algorithm Based on Hopfield Chaotic Neural Network," *IEEE Access*, vol. 7, pp. 185796 - 185810, 2019.

[12] Q. Lu, C. Zhu, and X. Deng, " An Efficient Image Encryption Scheme Based on the LSS Chaotic Map and Single S-Box," *IEEE Access*, vol. 8, pp.25664-25678, 2020.

[13] R. I. Abdelfattah, H. Mohamed, and M. E. Nasr, " Secure Image Encryption Scheme Based on DNA and New Multi Chaotic Map," in *4th international Conference on Advanced Technology and Applied Sciences*, Cairo, Egypt, 2019.