Defence Science Journal, Vol. 75, No. 1, January 2025, pp. 52-63, DOI : 10.14429/dsj.20107

Finite Field-Based Three-Tier Cryptography Algorithm to Secure the Images

M. Lavanya[#], K. Joseph Abraham Sundar[#,*] and S. Saravanan[$]

[#]*School of Computing, SASTRA Deemed University, Thanjavur - 613 401, India*
[$]*Department of Electronics and Communication Engineering, SASTRA Deemed University, Kumbakonam - 612 001, India*
[*]*E-mail: josephabrahamsundar@it.sastra.edu*

ABSTRACT

Securing the information is an important component in the computer network domain. Image information security is a vital part of the information security. The main process of image cryptography is traversing the image cryptosystem with high processing power, and efficiency. It is in terms of satisfying the cryptography requirements like confidentiality, integrity, and authenticity. A finite field-based image cryptography algorithm called TIEA (Three-tier Image Encryption Algorithm) is proposed in this paper. This algorithm deliberated Shannon's principles of cryptography like confusion and diffusion techniques of the images based on the finite field values. This paper also designed the key stream generation pattern based on the crypto key length. Two subkeys are generated for the purpose of crypto key and the key generation process is used to enhance the permutation of the key. Various benchmark images were tested with this proposed algorithm and also with other existing algorithms. The performance result shows that the proposed algorithm TIEA could be a better candidate algorithm for image security in the network domain.

Keywords: Shannon's principle; Finite field values; Image security; Confusion and diffusion

1. INTRODUCTION

In recent years the information security domain and image security has been a significant research topic. The images may be non-sensitive or very sensitive and healthcare also often transfers personal data[1]. These sensitive data may be prone to various attacks such as interception, fabrication, denial of service, and accessing data in an unauthorized manner. Thus, protecting the information during transmission is a vital process. Unlike the algorithms used to encrypt the text data, algorithms used to encrypt the image data require special features to satisfy the characteristics of image security processing. The existing algorithms like AES, DES, and various public key cryptography algorithms need to be combined with Cipher Block Chaining (CBC) to enhance the security level of image data[2].

Encryption of the images is based on the speed of the algorithm processing. Thus the study of image-based encryption algorithms is more required than the algorithms with fast processing. In[3] an elaborate survey was conducted on the classification of chaotic systems for image encryption algorithms. Numerous chaos-based algorithms were proposed[4-9]. A probabilistic symmetric encryption was proposed using a chaos scheme with suitable random bits in the insertion phase[10]. This method used 4 rounds of 2-staged diffusion which involves exclusive-OR operation. This method also increased the cipher text space and gave more resist to statistical attacks.

A hyper-chaotic system was used to sum the pixels along with different summation processes[11]. The NPCR and UACI were observed. A pathological image encryption method[12] used an external one-time keys method to validate the polynomial multiplication over a Galois field. The results were observed with look-up tables, avalanche effect, and encryption rounds.

Encrypting the image is a big exceptional task than encrypting the text data. The nearby pixel values in the image may have a high correlation and these values are used by the crypt analyzer to analyze the data easily. The generated cipher image must be very random, unpredictable and it should produce distributed histogram results and should satisfy all the statistical tests given by NIST[13]. A shuffling algorithm was proposed to leverage the pseudo-random sequences to enhance the performances of the initial S-Box and verify the image encryption scheme with various RGB color images[14]. Image encryption algorithm SIEA with lightweight processing methodology is shown in the paper [15]. The encryption procedure should be very sensitive and the minor change in the original image should produce a major impact on the cipher image.

In this paper, an encryption algorithm TIEA is proposed. This algorithm follows Shannon's confusion, diffusion logic and finite field in number theory logic. The rest of the paper is designed as follows. The proposed algorithms and the process are highlighted in the section 2. Implementation and results are given in section 3. Performance analysis is discussed in section 4.

2. PROPOSED ALGORITHM

In a network transmission, information security is a vital

Received : 16 April 2024, Revised : 10 September 2024
Accepted : 30 September 2024, Online published : 10 January 2025

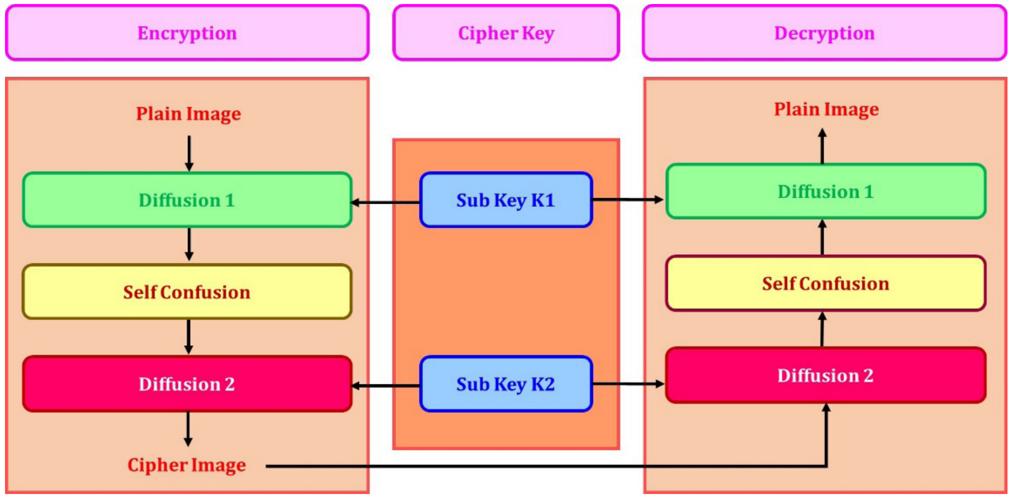


Figure 1. Flow diagram of the proposed algorithm.

process and the algorithm used to secure the data should satisfy the cryptography requirements such as confidentiality, integrity, and availability. It ensures the security of the image data using the proposed TIEA algorithm in this paper. This algorithm is a symmetric model and it has three phases such as key generation, encryption, and decryption. The finite field concept of number theory is used in this algorithm. Initially large prime number is chosen and the cipher key size is dynamically fixed based on the plain image row and column pixel size. Encryption and decryption processes are divided into three phases such as diffusion1, self-confusion, and diffusion 2. The flow diagram of the proposed algorithm is shown in Fig. 1 and the pseudo codes of the modules are given in algorithms 1-4.

2.1 Key Generation

Key generation plays an important role in cryptography. In this proposed algorithm two subkeys k1 and k2 are generated using the cipher key "K". Subkeys are generated as a matrix $K1_{(m,n)}$ and $K2_{(m,n)}$ where m and n are the row and column size of plain image pixel values. The size of K is equal to (m × n). Figure 2 shows the flow diagram of the key generation module. Subkeys k1 and k2 are generated based on the following steps.

- Step 1: Enter cipher key "K" as a input. Choose the random large prime number P
- Step2: Find the factors of the number 'r" where p-1 ≥ r ≥ 3 and form a group for the numbers [i]=F r ={f 1 ,f 2 ,…f K } where i ≥ 1
- Step3: Subkey K1 matrix is formed with G[1] to G[m] as a row seed values (rs)and G[r+1] to G[r+c+1] as a column seed values(cs).
- Step4: Multiplicative inverse of the row seed value by column seed value is calculated if both rs and cs are relatively prime numbers K1(r,c)=(rs -1 mod cs) where 1≤ r≤ m and where 1≤ c≤ n
- Step5: If rs and cs are not relatively prime then both seed values are added and mod with the P value. K1(r,c) =(rs+cs) mod P
- Step6: Subkey K2 matrix is formed with G[last] to G[last-m] as a row seed values (rs) and G[last-m-1] to G[last-m-1-n] as a column seed values(cs).
- Step7: Multiplicative inverse of the row seed value by column seed value is calculated if both rs and cs are relatively prime numbers. K2(r,c)=(rs -1 mod cs) where 1≤ r≤ m and where 1≤ c≤ n
- Step8: If rs and cs are not relatively prime then both seed values are added and mod with the P value. K2(r,c) =(rs+cs) mod P
- Step9: Find the position of the values in the cipher key "K" and place it in the subkey matrix "k1" to form a subkey k1 matrix. K1(r,c)=K[K1(r,c)]

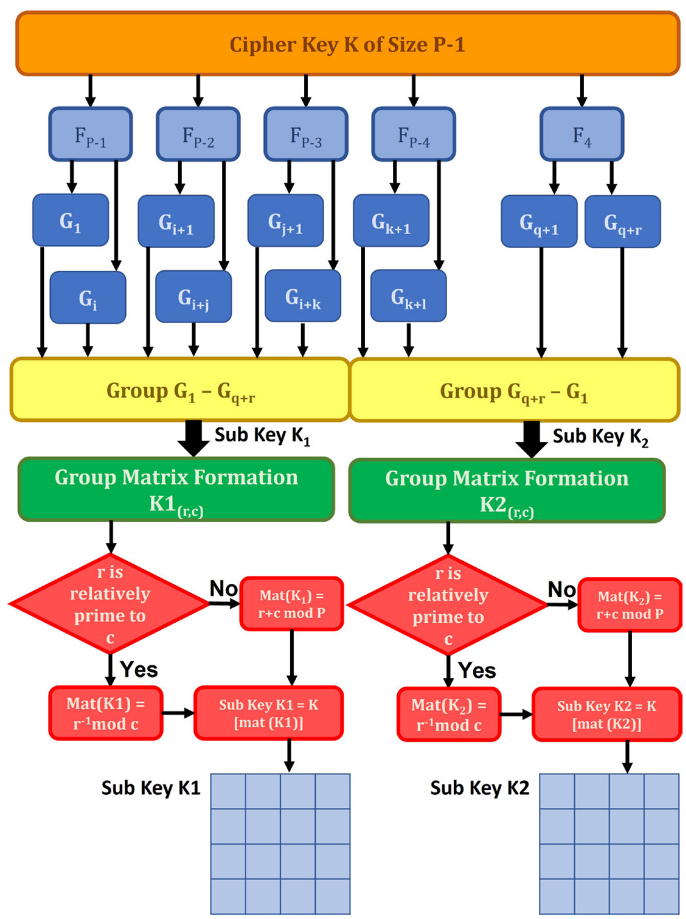


Figure 2. Key generation module.

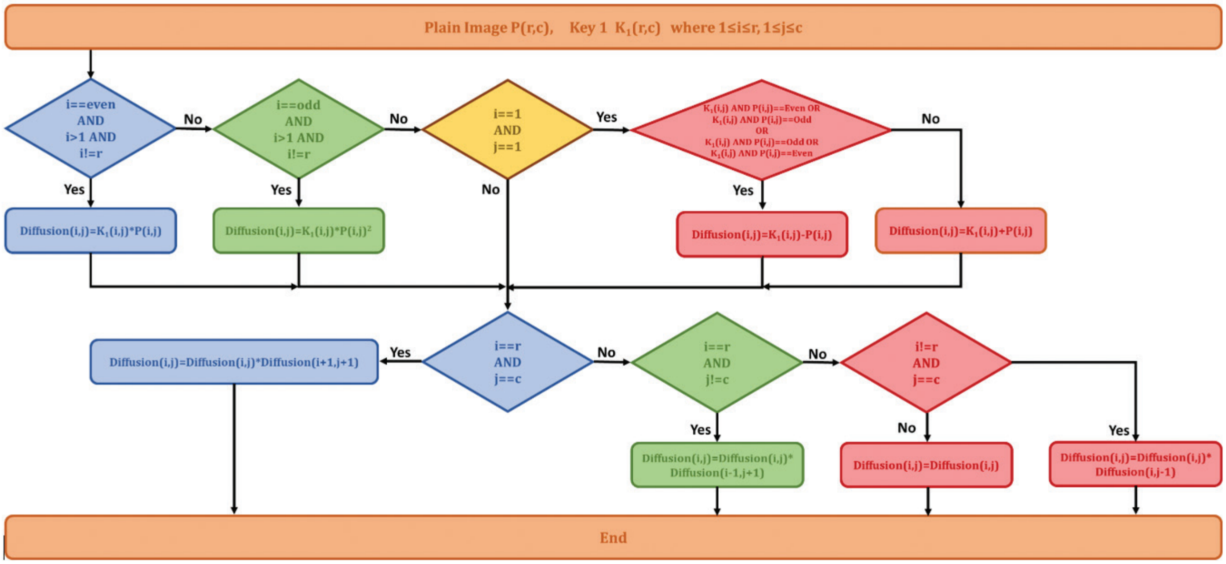


Figure 3. Flow diagram of Diffusion 1.

- Step10: Find the position of the values in the cipher key "K" and place it in the subkey matrix "k2" to form a subkey k2 matrix. K2(r,c)=K[K2(r,c)]

Algorithm 1. Key generation

```
Begin
i← p-1; k ← 1; m ← row; n ← Column;
    r←0 ; c ← 0; j ← 0;
    While i ≥ 4 do
      | Group_k ← Factor_(p-i)
      | k ← k+1
      | i=i+1
    End While
    While r! = m AND c == n do
      If GCD (Group_j, Group_(m+j)) ==1
          | K1_(r,c) ← Group_j MOD Group_(m+j)
          Else
          | K1_(r,c) ← (Group_j + Group_(m+j)) MOD P
      Endif
       j ← j +1
       c ← c+1
      If c==n AND r == m
         |c ← 0
         |r ← r+1
      Endif
End While

    r←0 ; c ← 0; j ← 1;
    While r != m AND c == n do
      If GCD (Group_(k-j), Group_(k-m-j) ) ==1
        | K2_(r,c) ← Group_(k-1) MOD Group_(k-m-1)
      Else
        | K2_(r,c) ← (Group_(k-1) + Group_(k-m-1) ) MOD p
      Endif
        j ← j +1
        c ← c+1
      If c==n AND r == m
        |c ← 0
        |r ← r+1
 Endif
 End While
 End
```

2.2 Encryption

The second module is the encryption module. Plain image is encrypted with the subkey values K1 and K2 based on Shannon's principal diffusion and confusion logic. Encryption process of proposed algorithm in divided in to three modules named as Key1 diffusion, Self-confusion and key2 diffusion. Plain image pixel values are process with K1 and K2 in diffusion modules and in self-confusion no key values are involved.

2.2.1 Key Diffusion 1

Flow diagram of diffusion1 module is given in the Fig. 3.

Algorithm 2. Diffusion 1

```
Begin
   i ← 1; j ← 1; m ← row; n ← column
While i ≤ m do
 | While j ≤ n do
 |     If i==1 AND j==1
 |       If(K1_(i,j) %2==0 AND PI_(i,j) %2==0)OR(K1_(i,j) %2==1 AND PI_(i,j) %2==1)
 |         | Diffusion1_(i,j) ← K1_(i,j) + PI_(i,j)
 |       Else if(K1_(i,j) mod 2==0 AND PI_(i,j) %2==1) OR K1_(i,j) %2==1)AND PI_(i,j)
 |           %2==0)
 |         | Diffusion1_(i,j) ← K1_(i,j) -PI_(i,j)
 |       Endif
 |     Elseif i==m AND j==n
 |        If(K1_(i,j) % 2==0 AND PI_(i,j) %2==0)  OR (K1_(i,j) %2==1)AND PI_(i,j)
 |         | %2==1)
 |         | Diffusion1_(i,j) ← K1_(i,j) -PI_(i,j)
 |        Elseif(K1_(i,j)  %2==0 AND PI_(i,j) %2==1) OR K1_(i,j) %2==1)AND PI_(i,j)
 |         | %2==0)
 |         | Diffusion1_(i,j) ← K1_(i,j) +PI_(i,j)
 |        Endif
 |      Endif
 |      If(I %2 == 1)
 |        |Diffusion1 = K1_(i,j) x PI_(i,j)
 |      Elseif(i %2 == 0)
 |        |Diffusion1 = K1_(i,j) x PI_(i,j)
 |      Endif
 |       If (i==m AND j==n)
 |        | FinalDiff1_(i,j) = Diffusion1_(i,j) x Diffusion1_(i+1,j+1)
 |       Elseif(i==m AND j!=n)
 |        | FinalDiff1_(i,j) = Diffusion1_(i,j) x Diffusion1_(i-1,j+1)
 |        Elseif(i==m AND j==n)
 |        | FinalDiff1_(i,j) = Diffusion1_(i,j) x Diffusion1_(i,j-1)
 |        Endif
 |       j=j+1
 |   End while
 |   i=i+1
 End while
End
```

2.2.2 Self-Confusion

Diffusion 1 processed to find the final matrix value which involves plain image matrix and K1 matrix. The output of the diffusion1 module is processed using confusion logic. Flow diagram of self-confusion module is shown in Fig. 4. The final matrix FD1(mxn) follows the following rules to convert the diffused final matrix into self-confused matrix SC(m,n). Number keys are involved in this module.

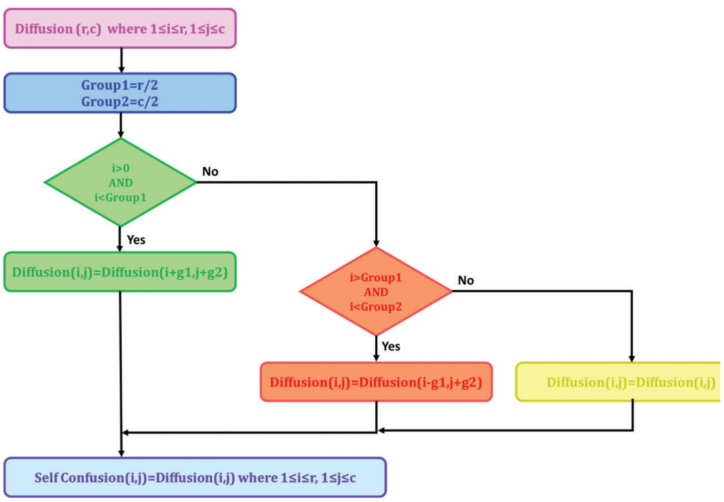


Figure 4. Flow diagram of self-confusion module.

Algorithm 3. Self-Confusion

Begin
$i \leftarrow 1; j \leftarrow 1; m \leftarrow row; n \leftarrow column$
$g1 \leftarrow m/2; g2 \leftarrow n/2$
While $i < m$ **do**
While $j < n$ **do**
$Finaldiff1_{(i,j)} \leftarrow Finaldiff1_{(i+g1),(j+g2)}$ where $0 < i < g1$
$Finaldiff1_{(i,j)} \leftarrow Finaldiff1_{(i-g1),(j+g2)}$ where $g1 < i < g2$
$selfconfusion_{(i,j)} \leftarrow Finaldiff1_{(i,j)}$
End While
j=j+1
End While
i=i+1
End

2.2.3 Key Diffusion 2

- *Step 1:* Self Confusion matrix value SC(mxn) is given as input
- *Step 2:* K2 matrix is converted from hexa to decimal values
- *Step 3:* Calculate the relative prime matrix RP(mxn) by following rules:
- *Step 3.1:* Find the list of relative prime numbers of the value K2(i,j) where 0<i<m+1
- and 0< j <n+1
- *Step 3.2:* Find the mean relative prime value from the list of relative prime numbers of K2(i,j) where 0<i<m+1 and 0< j <n+1
- *Step 4:* Find the modulo inverse matrix MD(m x n) using following equation
- $MD(i,j) = K2(i,j)^{-1} \bmod RP(i,j)$ where 0<i<m+1 and 0< j <n+1
- *Step 5:* Calculate the cipher image matric CM(mxn) multiplying Modulo inverse matrix and Cipher image matrix $CM(i,j) = (MD(i,j) \times SC(i,j)) \bmod 255$.

Flow diagram of diffusion 2 is shown in Fig. 5.

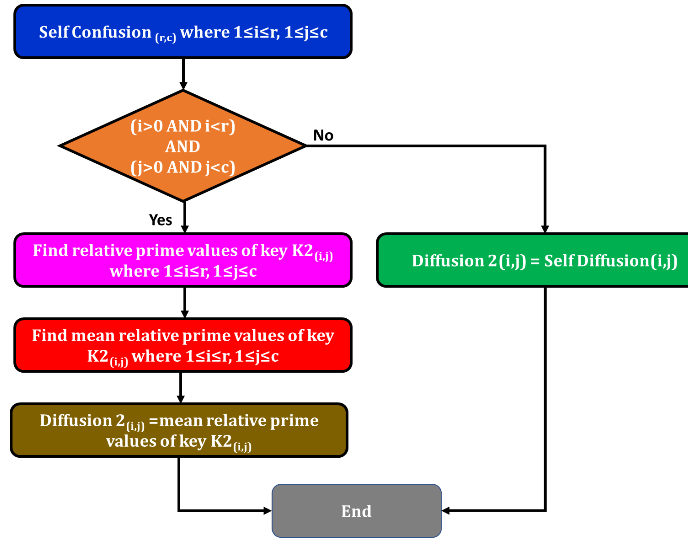


Figure 5. Flow diagram of Diffusion 2.

Algorithm 4: Diffusion 2

Begin
$i \leftarrow 1; j \leftarrow 1; m \leftarrow row; n \leftarrow column; y \leftarrow 0$
While $i \leq m$ **do**
While $j \leq n$ **do**
$Rg_y = K2_{(i,j)} \bmod P$
$y \leftarrow y+1$
$Mean = ½ \sum Rg_x$
$RP_{(I,j)} \leftarrow Mean$
$j=j+1$
End while
$i=i+1$
End while
$MD_{(I,j)} \leftarrow KR_{(I,j)} \bmod RP_{(I,j)}$ where $0<i \leq m, 0<j \leq n$
$CM_{(I,j)} \leftarrow MD_{(I,j)} \ X\ SC_{(I,j)} \bmod 255$ where $0<i \leq m, 0<j \leq n$
End

2.3 Decryption

The third module of this algorithm is the decryption process. Cipher image is decrypted with the subkey values K1 and K2 based on Shannon's principal diffusion and confusion logic. Decryption process of proposed algorithm in divided in to three modules named as Key2 diffusion, Self-confusion and key diffusion1. Cipher image pixel values are process with K1 and K2 in diffusion modules and in self-confusion no key values are involved.

2.3.1 Key Diffusion 2

- *Step 1:* Cipher image matrix value CM(4×4) is given as input
- *Step 2:* K2 matrix is converted from hex to decimal values
- *Step 3:* Calculate the relative prime matrix RP(4×4) by following rules:
- *Step 3.1:* Find the list of relative prime numbers of the value K2(i,j) where 0<i<m+1, and 0< j <n+1
- *Step 3.2:* Find the mean relative prime value from the list

of relative prime numbers of K2(i,j) where $0<i<m+1$ and $0<j<n+1$

- *Step 4:* Find the modulo inverse matrix MI(4×4) using following equation
  $MD(i,j) = K2(i,j)^{-1} \bmod RP(i,j)$ where $0<i<m+1$ and $0<j<n+1$
- *Step 5:* Calculate the Self-Confusion matrix SC(i,j) using following Eqn.
  $SC(i,j) = MI(i,j)^{-1} \times CM(i,j)$

2.3.2 Self-Confusion

Diffusion 2 processed to find the self-confusion matrix SC(i,j) which involves cipher image matrix and K2 matrix. The output of the diffusion2 module is processed using confusion logic. The SC matrix follows the following rules to convert the self-confusion matrix into final diffusion 1 matrix. No keys are involved in this module.

- *Step 1:* Self Confusion matrix SC(mxn) is divided based on the calculated value g1 and g2 Where the $g1 = m/2$ and $g2 = n/2$
- *Step 2:* Swap the values in SC matrix based on following rules:
  $SC(i,j) = SC((i+g1)(j+g2))$ where $0<i<g1$
  $SC(i,j) = SC((i-g1)(j+g2))$ where $g1<i<g2$
- *Step 3*: FD1(i,j)=SC(i,j)

2.3.3 Key Diffusion 1

- Step 1: Diffusion1 final matrix FD1(m × n) is calculated using following rules:
- Step 1.1: $D1(i,j) = FD1(i,j) \times FD1((i+1)(j+1))^{-1}$ where i=m and j =n
- Step 1.2: $D1(i,j) = FD1(i,j) \times FD1((i-1)(j+1))^{-1}$ where i=m and j !=n
- Step 1.3: $D1(i,j) = FD1(i,j) \times FD1((i)(j-1))^{-1}$ where i!=m and j =n
- Step 2: Diffusion1 matrix D1(mxn) is calculated based on following rules
- Step 2.1: if K1(i,j) mod 2 =0 AND D1(i,j) mod 2 = 0 OR K1(i,j) mod 2 =1 AND Pt(i,j) mod 2 = 1, Pt(i,j) = D1(i,j) - K1(i,j) + where i = 1 and j = 1
- Step 2.2: if K1(i,j) mod 2 =0 AND D1(i,j) mod 2 = 1OR K1(i,j) mod 2 =1 AND Pt(i,j) mod 2 =0, Pt(i,j) = D1(i,j) + K1(i,j) where i = 1 and j = 1
- Step2.3: if K1(i,j) mod 2 =0 AND D1(i,j) mod 2 = 0 OR K1(i,j) mod 2 =1 AND Pt(i,j) mod 2 = 1, Pt(i,j) =D1(i,j) + K1(i,j) where i =m and j = n
- Step 2.4: if K1(i,j) mod 2 =0 AND Pt(i,j) mod 2 = 1OR K1(i,j) mod 2 =1 AND Pt(i,j) mod 2 =0 Pt(i,j) = D1(i,j) - K1(i,j) where i =m and j = n
- Step 2.5: ifi mod 2 =1 $Pt(i,j) = D1(i,j) \times K1(i,j)^{-1}$ where $1<i<m$ and $1<j<n$
- Step 2.6: if i mod 2 =0 $Pt(i,j) = (D1(i,j) \times 2^{-1}) * K1(i,j)^{-1}$ where $1<i<m$ and $1<j<n$

3. IMPLEMENTATION AND EXPERIMENTAL RESULTS

The proposed TIEA algorithm modules are implemented

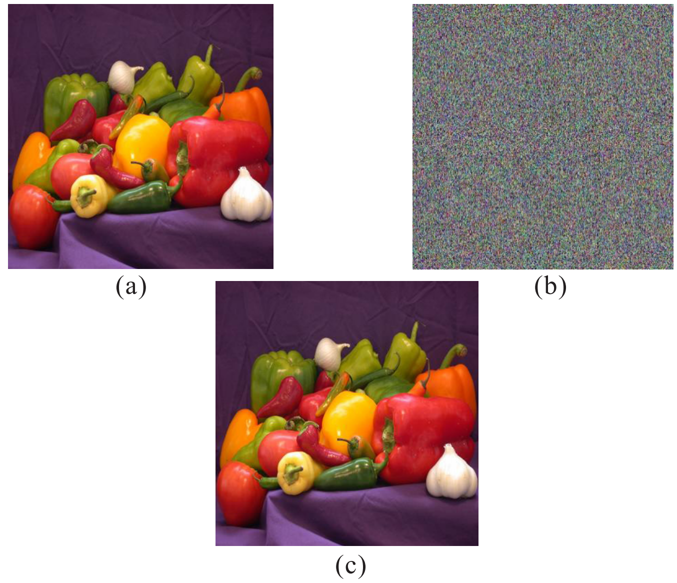


Figure 6. Color image vegetables; (a) Original image; (b) Encryption image; and (c) Decryption image.

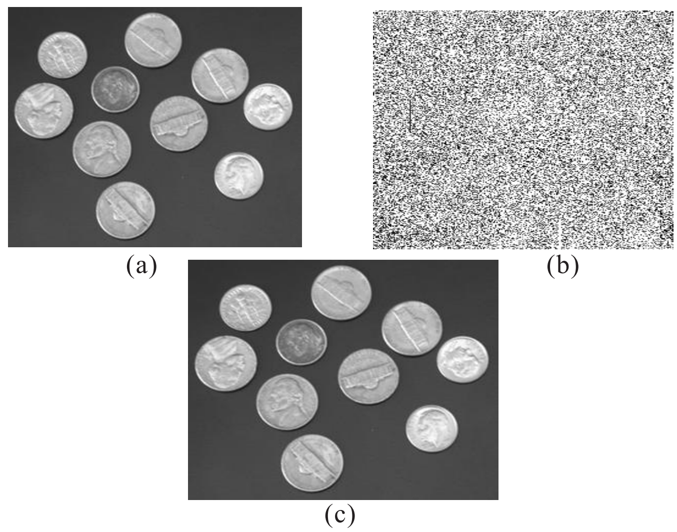


Figure 7. Dollar image; (a) Original image; (b) Encryption image; and (c) Decryption image.

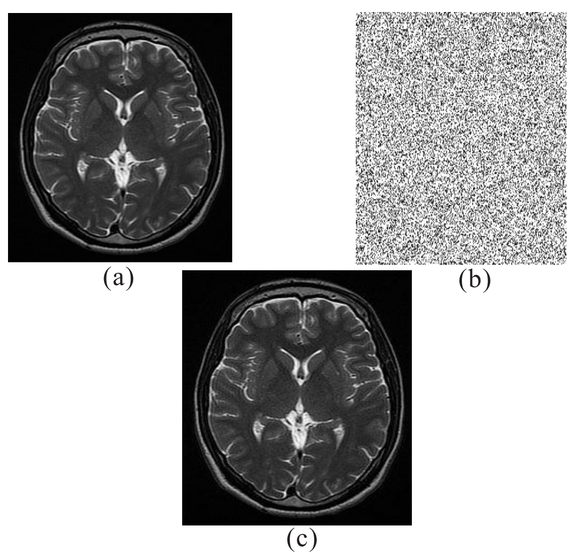


Figure 8. CT image Harns; (a) Original image; (b) Encryption image; and (c) Decryption image.

on MATLAB 2019b software with core i3, 2GB graphics card, and 8 GB RAM. Three bench mark images are taken from the MATLAB database of different scales such as gray, RGB color and medical image Harns Computed Tomography (CT)[18-20] involved in the testing process of the algorithm. Different dimension images are used to check the scalability of the algorithm. Figure 6 to Fig. 8 shows the plain image and cipher image of bench mark data.

4. PERFORMANCE ANALYSIS

To verify and prove the achievement ability and the security level of the proposed TIEA algorithm numerous trails have been processed to validate the qualitative and quantitative measures. The proposed algorithm is resistant to the exhaustive search analyze and it strictly follows diffusion and confusion technique. Existing DES algorithm key size is 56 bit and AES key size is 128 bits fixed. Compare with these existing algorithms the TIEA algorithm key size is dynamic in nature based on the image pixel size. Proposed TIEA algorithm was compared with the standard algorithms AES[16] and holomorphic encryption[17]. Numerous procedures are followed to perform the analyzation of the algorithms such as Differential analysis, Correlation analysis, Histogram analysis, key sensitivity analysis[19] and the results and cipher outputs shows the randomness of the encrypted images.

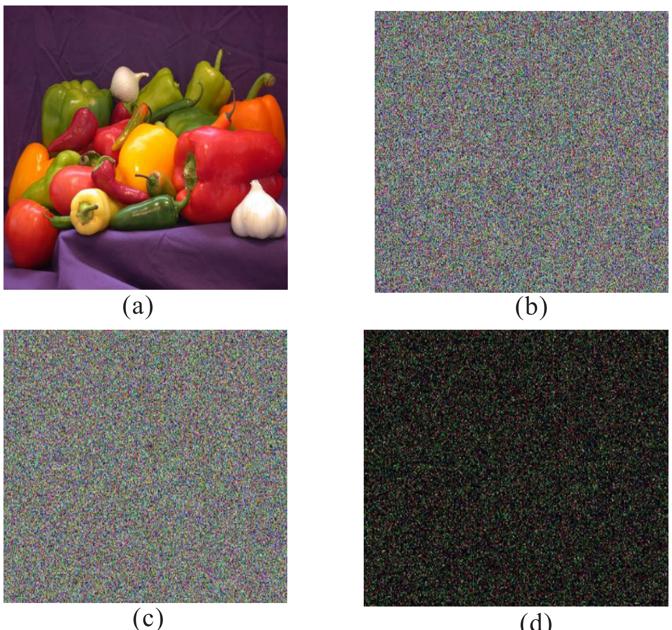


Figure 9. Color image vegetables; (a) Original image; (b) Encryption 1; (c) Encryption 2; and (d) Difference between 12a and 12b.

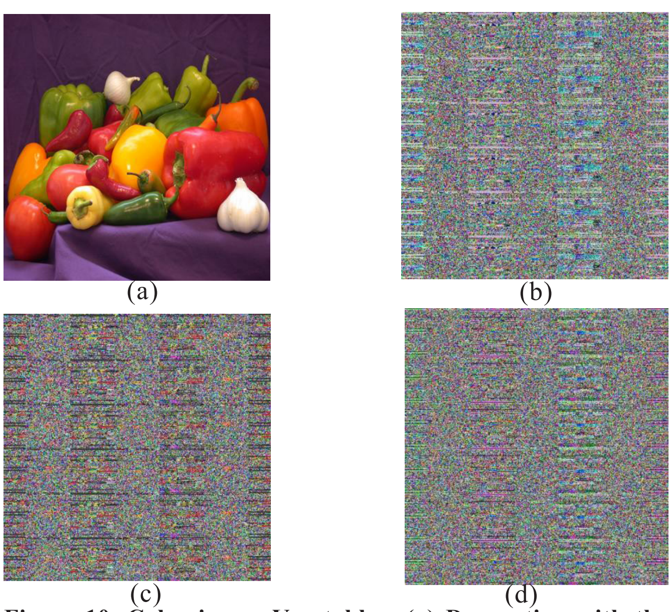


Figure 10. Color image Vegetables; (a) Decryption with the correct key; (b) Decryption with a 2-bit modified Key; (c) Decryption with a 4-bit modified Key; and (d) Decryption with 8-bit modified Key.

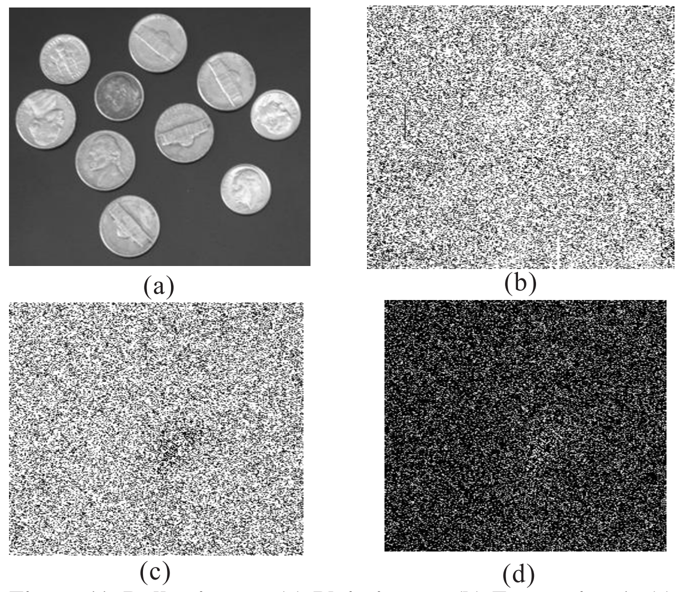


Figure 11. Dollar image; (a) Plain image; (b) Encryption 1; (c) Encryption 2; and (d) Difference between 14b and 14c.

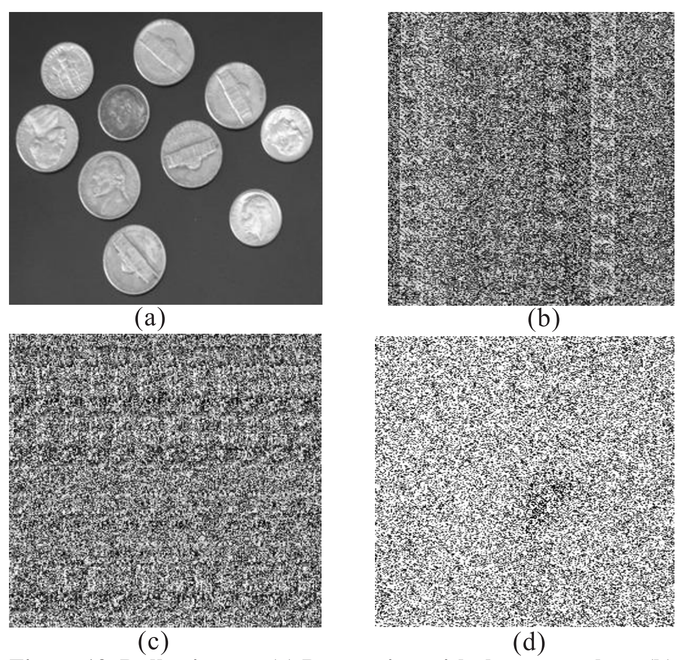


Figure 12. Dollar image; (a) Decryption with the correct key; (b) Decryption with a 2-bit modified key; (c) Decryption with a 4-bit modified key; and (d) Decryption with an 8-bit modified key.

4.1 Key Sensitivity Analysis

The key sensitivity analysis is the vital procedure to validate the proposed algorithm in terms of the randomness of the results with respect to the key and the avalanche effect of the algorithm. Various scale images are considered to test and perform key analyzation. Cipher key is slightly modified and the sub keys K1 and K2 are generated to test the cipher image randomness. The figure 9-14 Shows and prove that the proposed algorithm provides the random outputs and there is no similarity between the cipher images. For each and every different key the random cipher image is generated. Hence it is proved that the proposed TIEA algorithm performs well in terms of key sensitivity and provides the adequate security to the image data while transmission.

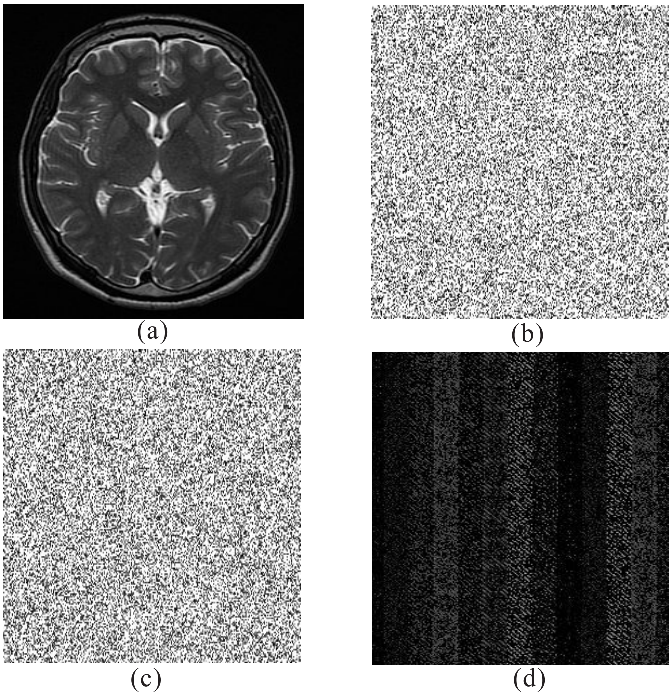


Figure 13. CT image Harns; (a) Plain image; (b) Encryption 1; (c) Encryption 2; and (d) Difference between 16b and 16c.

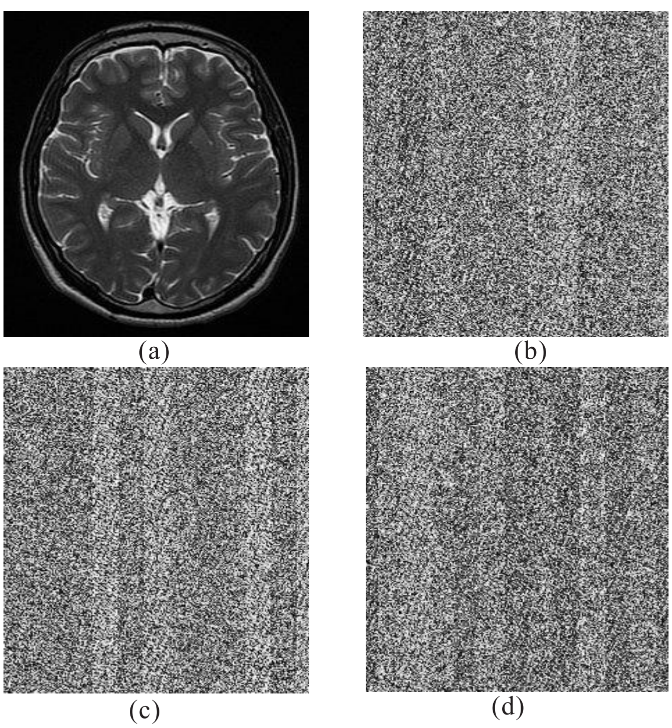


Figure 14. CT image Harns; (a) Decryption with the correct key; (b) Decryption with a 2-bit modified key; (c) Decryption with a 4-bit modified key; and (d) Decryption with 8-bit modified key.

4.2 Histogram Analysis

Plain images and encrypted images are differentiated with the pixel values. The pixel value positions of these images are examined and verified using the histogram analysis. Pixel values of the original image are in non-uniform and random positions in histograms[20]. To overcome the statistical attack the position of pixel values in cipher image is very important. Image balancing and placing the pixels in distributed and decentralized manner is essential to prove the randomness of the pixel positions. The pixel values are uniformly distributed in the histogram analysis diagram shown in Fig. 15- Fig. 17.

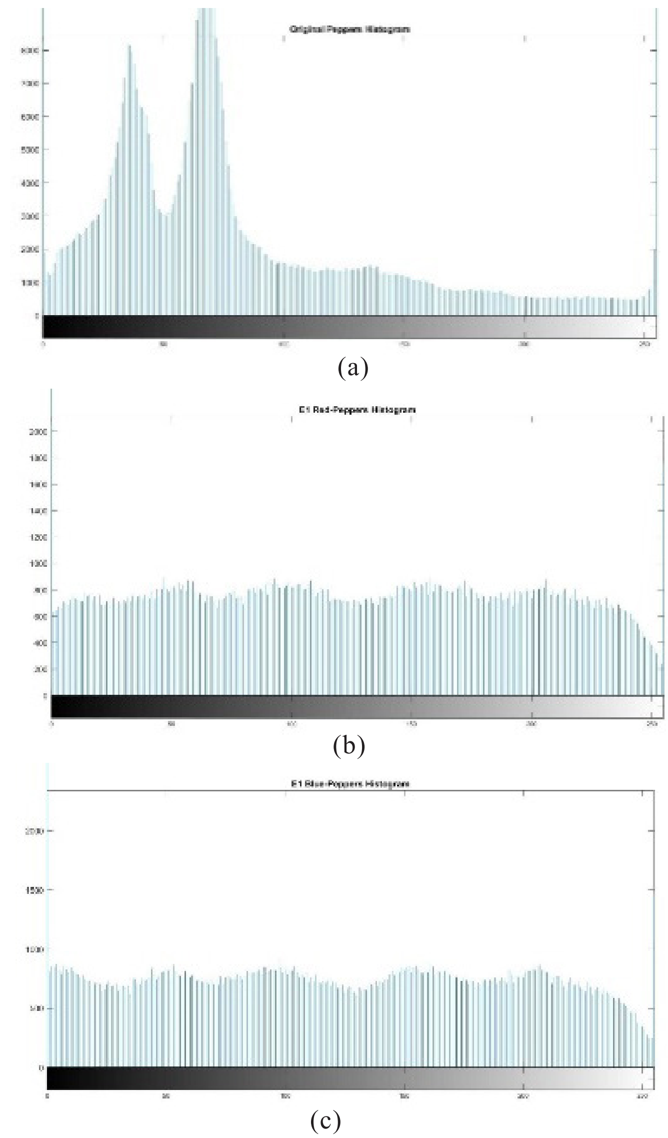


Figure 15. Histogram; (a) Vegetables plain image; (b) Encryption 1; and (c) Encryption 2.

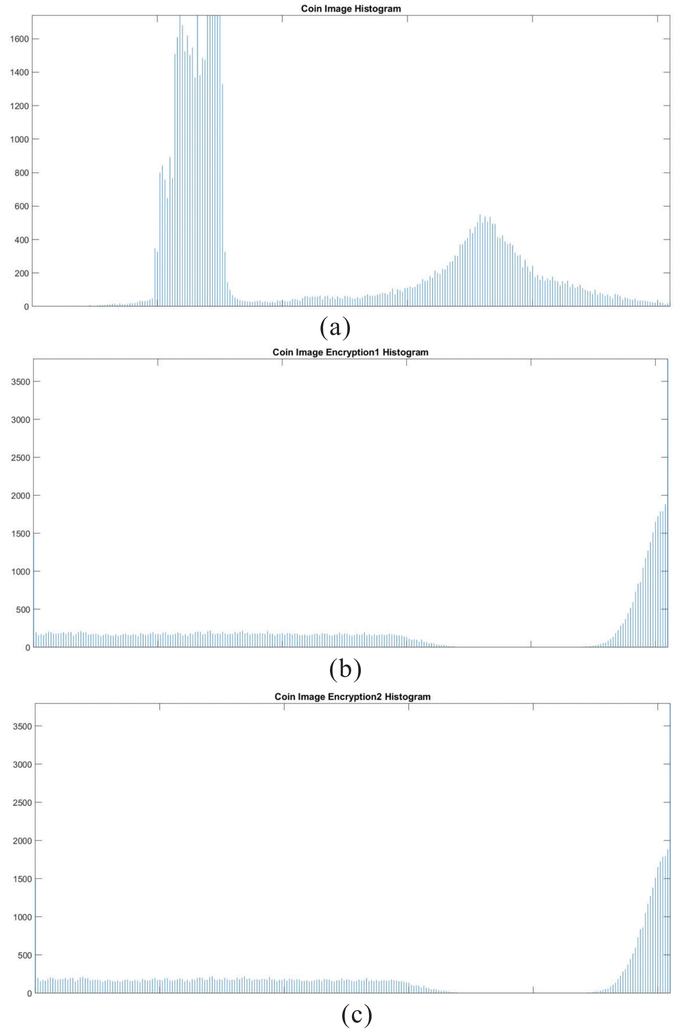


Figure 16. Histogram; (a) Dollar plain image; (b) Encryption 1; and (c) Encryption 2.

4.3 Correlation of Adjacent Pixels

The plain image pixel values have the high and close correlation with the neighboring pixel values. These high correlation increases the chances of statistical attack by the analyst. Hence the encryption process focuses on reducing the correlation values among the neighboring pixel values in the encrypted image to reduce the possibilities of the attacks. Eqn. 1 shows the correlation coefficient values of the encrypted image.

$$
\begin{aligned}
ek(i) &= \frac{1}{n}\sum\nolimits_{l=1}^{n} i_l \\
dk(i) &= \frac{1}{n}\sum\nolimits_{l=1}^{n} (i_l - ek(i))^2 \qquad (1) \\
c(i,j) &= \frac{1}{n}\sum\nolimits_{l=1}^{n} (i_l - ek(i)) - (j_l - ek(j))
\end{aligned}
$$

Gray measurements of two nearby pixels are denoted by i and j values. Figure 18 - Fig. 24, shows the outputs of the correlation values of the plain and cipher images with respect to the coefficient values of Horizontal (H), Diagonal (D) and vertical (V). The output figures shows that the correlation values are decreased in the cipher images compared with the plain images. The correlation analysis output shown is shown in the Table 1.

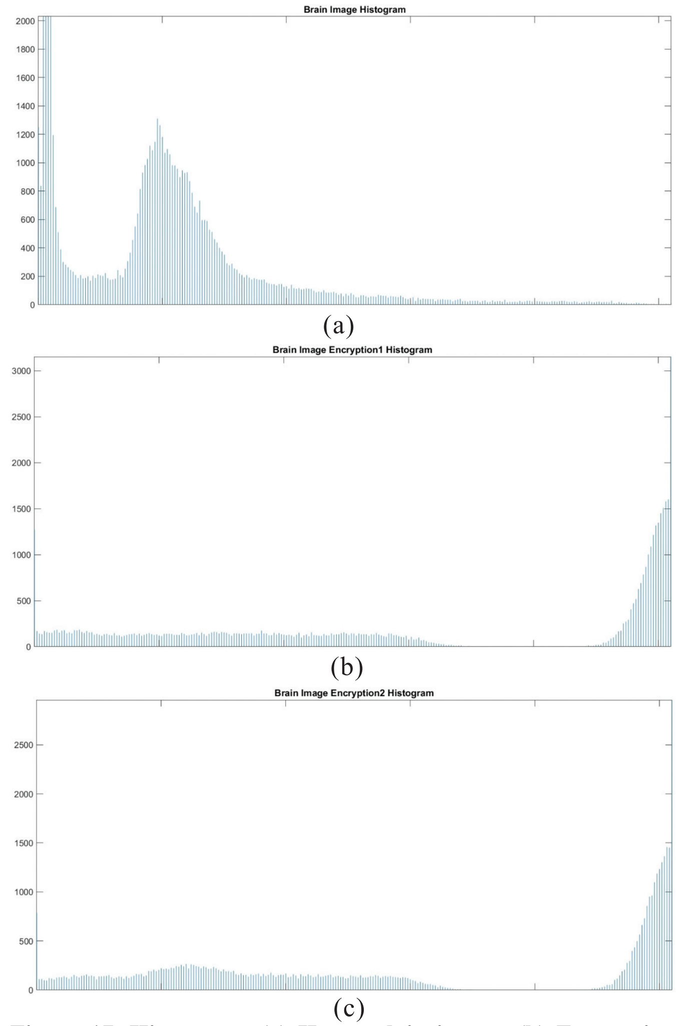


Figure 17. Histogram; (a) Harns plain image; (b) Encryption 1; and (c) Encryption 2.

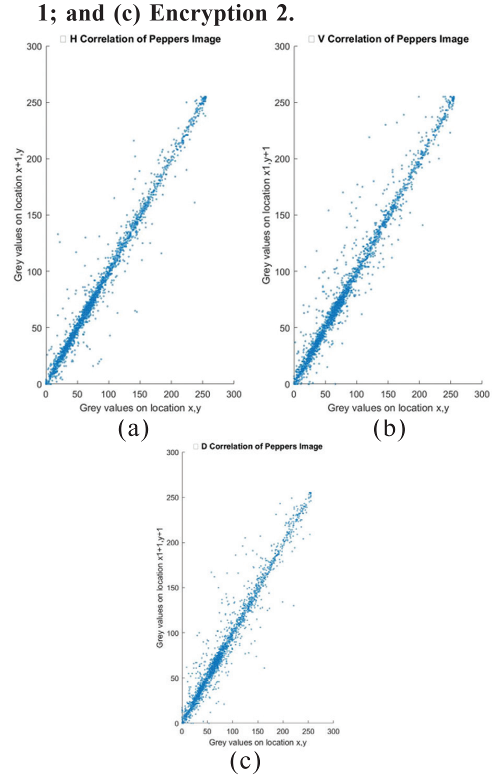


Figure 18. Vegetables plain Image; (a) H_Correlation; (b) V_Correlation; and (c) D_Correlation.

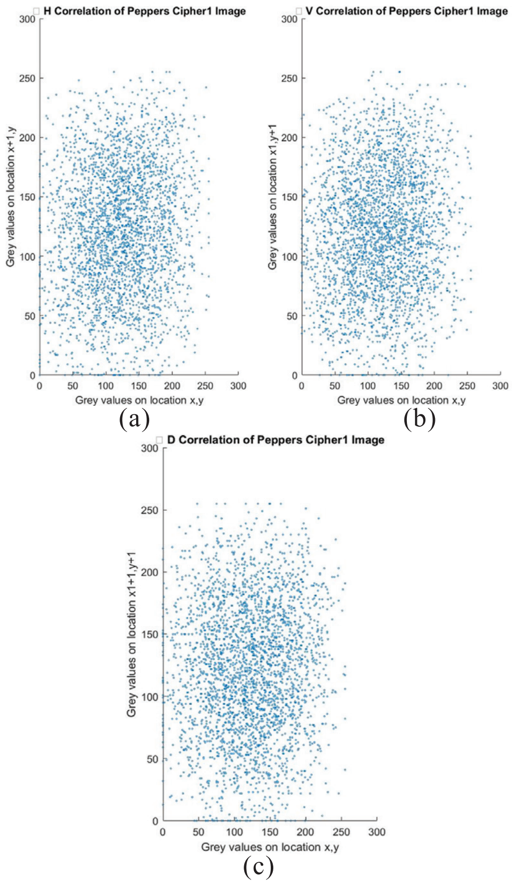


Figure 19. Vegetables cipher Image; (a) H_Correlation; (b) V_Correlation; and (c) D_Correlation.

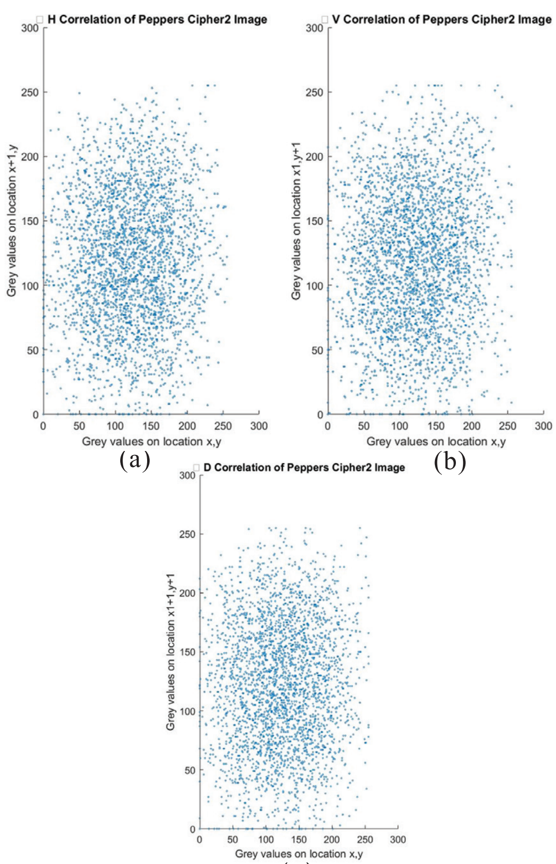


Figure 20. Vegetables cipher image 2; (a) H_Correlation; (b) V_Correlation; and (c) D_Correlation.

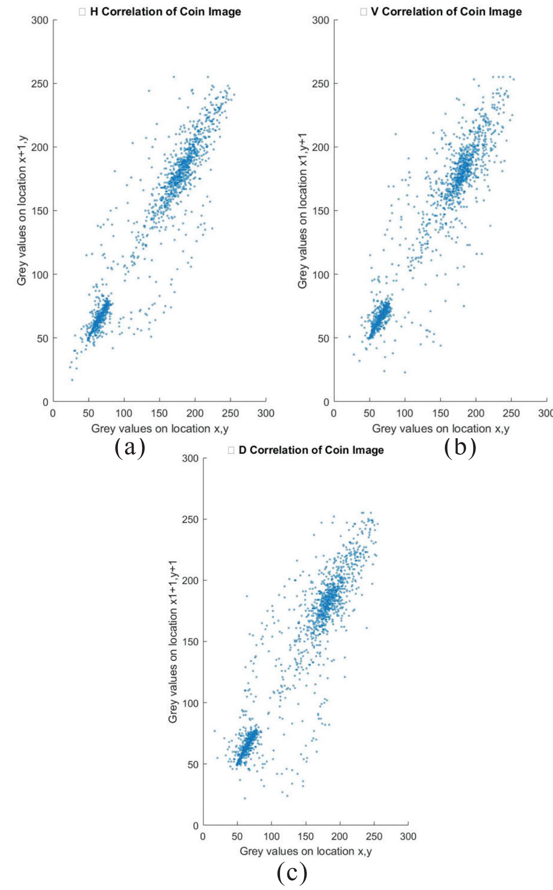


Figure 21. Dollar plain image; (a) H_Correlation; (b) V_ Correlation; and (c) D_Correlation.

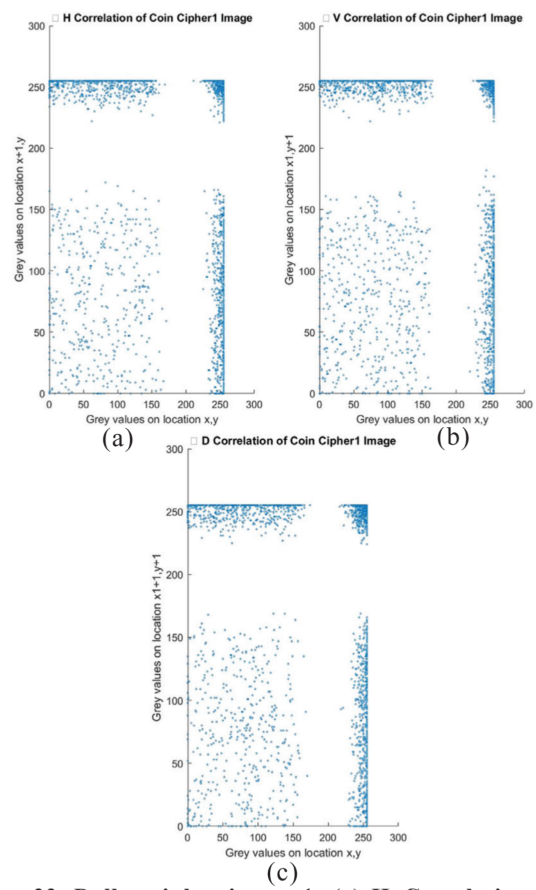
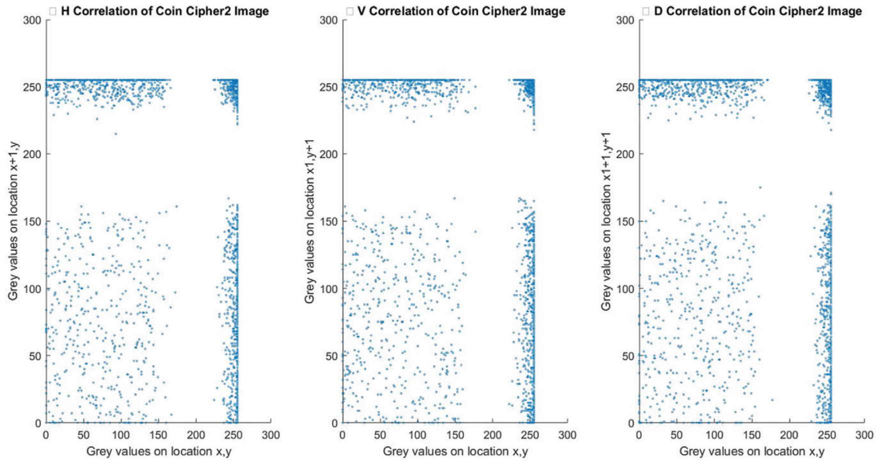
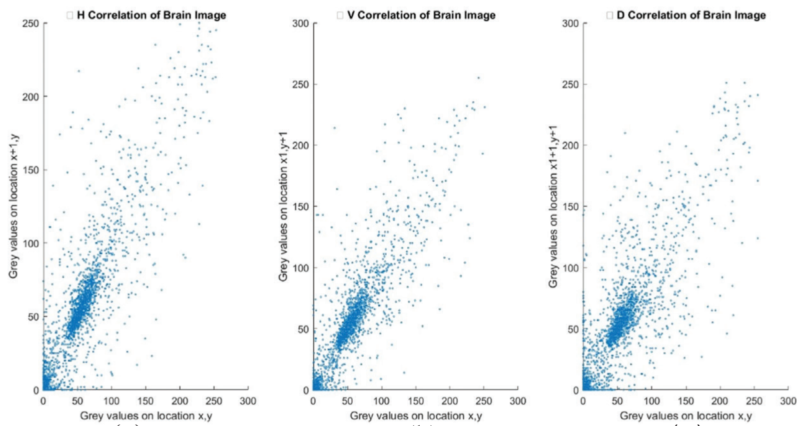


Figure 22. Dollar cipher image 1; (a) H_Correlation; (b) V_ Correlation; and (c) D_Correlation.



(a) (b) (c)

Figure 23. Dollar cipher image 2; (a) H_Correlation; (b) V_Correlation; and (c) D_Correlation.



(a) (b) (c)

Figure 24. Harns plain image; (a) H_Correlation; (b) V_Correlation; and (c) D_Correlation.

Table 1. Results of correlation analysis

Image	Vegetables (512 x 512) color image			Dollar (1024 x 1024) gray scale image			Harns (220 x 275) CT image		
	Plain	**Cipher 1**	**Cipher 2**	**Plain**	**Cipher 1**	**Cipher 2**	**Plain**	**Cipher 1**	**Cipher 2**
Horizontal	0.9922	0.0510	0.2486	0.9752	-0.0056	-0.0292	0.8997	-0.0324	-0.0088
Vertical	0.9906	0.1166	0.2465	0.9716	0.0202	-0.0213	0.9078	-0.0096	-0.0023
Diagonal	0.9819	0.0763	0.4672	0.9571	0.0468	0.0165	0.8338	-0.0191	-0.0199

4.4 Information Entropy Analysis

The qualitative measures the cipher image randomness. The information entropy is used to find the randomness of the encrypted image. Eqn. 2 shows the formula used to calculate the information entropy value.

$$IE(r) = \sum_{i=0}^{2^n-1} l(r_x) \log_l \frac{1}{r_i} \tag{2}$$

The images have 8 as an entropy value. Table 2 shows the information entropy values of the encrypted images. The values of the cipher images are close to their plain image entropy values that show the pixel loss of the cipher image is reduced and the Table 3 shows the efficiency of the proposed algorithm with respect to the information entropy.

Table 2. Information entropy

Image	Information entropy
Vegetables_Original	7.37
Vegetables_Encryption1	7.73
Vegetables_Encryption2	7.74
Dollar_Original	6.33
Dollar_Encryption1	5.66
Dollar_Encryption2	5.68
Harns_ Original	6.27
Harns_Encryption1	5.59

4.5 Differential Attack Analysis

The efficient cryptography algorithms have the features that the cipher images pixel values are sensitive to the plain images. Minor changes in the plain images must make the major changes in the cipher images in the efficient algorithm. The Unified Average Change Intensity (UACI) and Pixel Change Rate (NPCR) are the two parameters used in the differential analysis to identify and prove the efficiency of the proposed

Table 3. Differential attack analysis

Image	NPCR_Score	NPCR_dist	UACI_Score	UACI_dist
Vegetables	0.9941	0.9961	0.2317	0.3347
Dollar	0.8916	0.9961	0.3772	0.3346
Harns	0.8944	0.9961	0.3728	0.3346

Table 4. Comparison of the proposed method with other methods

Algorithm	Correlation analysis			Information entropy	Differential attack analysis	
	H	V	D		NPCR	UACI
M-AES[17]	-0.0039	0.0058	0.0023	6.5653	99.59	31.06
Holomorphic encryption[17]	-0.0007	0.0029	0.0020	6.5791	99.60	31.13
SIEA[15]	0.02728	0.03793	0.0709	6.6919	99.61	33.46
Proposed TIEA	-0.0347	-0.0231	-0.0109	7.00	99.61	33.46

algorithm in terms of the differential analysis. Eqn. 3 and Eqn. 4 shows the formula to find the NPCR and the UACI values.

$$N = \frac{\sum dk(i,j)}{m*n} \tag{3}$$

$$U = \frac{1}{m*n}\left(\sum \frac{ek1(i,j) - ek2(i,j)}{255}\right) \tag{4}$$

Plain images are encrypted with two different keys and the ek1 and ek1 are the different encrypted images. The row and column values are denoted by m and n. The results must be close to 1 to prove the efficiency of the algorithms. The proposed algorithm is resistant to the differential attack shown in the table 3. Quantitative measures are defined by the NPCR Score with almost 1 is good. Qualitative measure is calculated with the NPCR_pval value and Mean average is denoted with the NPCR_dist. In the other end UACI score should give very low value close to 0 and mean of UACI is denoted by UACI_dist.

The proposed TIEA algorithm performance is compared with M-AES[1], Holomorphic Encryption[17] and SIEA[15] and the results were tabulated in Table 4. The values in the table gives the clear ideas that the proposed algorithm performance is better and efficient compare with the existing algorithms. The TIEA algorithm is designed with the finite field concepts in the number theory and the dynamic cipher key generation model. Also, the algorithm processing time is less and the light weight procedures are followed compare with the standard algorithms.

5. CONCLUSION

The proposed TIEA algorithm is used to transfer the image data securely. During transmission, sensitive images such as medical-related scan images or X-Ray images security is very essential and data loss also to be reduced to assure the receiver that he received the correct image. To ensure the correctness of the image and to increase the randomness of the cipher image the proposed algorithm is designed efficiently. The algorithm performance is tested with various methods with variations in the input images such as black and white, grayscale, color, and CT Images. The algorithm follows the confusion and diffusion techniques to increase the randomness and complexity of an algorithm. The experimental results show that the proposed algorithm is performing well with various dimensions of images. The cipher key is given by the sender with dynamic size and the subkeys K1 and K2 are generated with fixed size which helps the encryption process to become lightweight. The complexity of the proposed algorithm is $O(n^2)$. The performance analysis shows the benefits of the proposed TIEA algorithm. Even though it is proposed for image encryption still the text data can be used by the sender to send the data securely.

REFERENCES

1. Alawida, M.; Samsudin, A. & Teh, J.S. A new hybrid digital chaotic system with applications in image encryption. *Signal Process*., 2019, **160**, 45–58. doi:10.1016/j.sigpro.2019.02.016
2. Paar, C. & Pelzl, J. Understanding cryptography: A textbook for students and practitioners. Springer, Heidelberg, 2009, 292p.
3. Pengfei, Fang.; Han, Liu. & Chengmao, Wu. A survey of image encryption algorithms based on chaotic system. *Vis Comput*., 2023, **39**, 1975–2003. doi:10.1007/s00371-022-02459-5
4. Zahmoul, R.; Ejbali, R. & Zaied, M. Image encryption based on new beta chaotic maps, *Opt. Lasers Eng*., 2017, **96**, 39-49. doi:10.1016/j.optlaseng.2017.04.009
5. Liu, H. & Wang, X. Colour image encryption using spatial bit-level permutation and high-dimension chaotic system. *Opt. Commun*. 2011, **284**(16), 3895–390. doi:10.1016/j.optcom.2011.04.001
6. Guan, Z.H.; Huang, F. & Guan, W. Chaos-based image encryption algorithm. *Phys. Lett. A*. 2005, **346**(1), 153–157. doi:10.1016/j.physleta.2005.08.006
7. Norouzi, B.; Mirzakuchaki, S.; Seyedzadeh, S.M. & Mosavi, M.R. A simple, sensitive and secure image encryption algorithm based on a hyper-chaotic system with only one round diffusion process. *Multimedia Tools App*., 2012, **71**(3), 1469–1497. doi:10.1007/s11042-012-1292-9

8. Guodong, Ye. & Kwok-Wo, Wong. An efficient chaotic image encryption algorithm based on a generalised Arnold map. *Nonlinear Dyn*. 2012, **69**(4), 2079–2087. doi:10.1007/s11071-012-0409-z
9. Alireza, Arab.; Mohammad, Javad, Rostami. & Behnam, Ghavami. An image encryption method based on chaos system and AES algorithm. *J. Supercomput*. 2019, **75**(10), 6663–6682. doi:10.1007/s11227-019-02878-7
10. Dhall, S.; Pal, S.K. & Sharma, K. A chaos-based probabilistic block cipher for image encryption. *J. King Saud Univ. - Comput. Inf. Sci.,* 2022, **34**(1), 1533-1544. doi:10.1016/j.jksuci.2018.09.015
11. Norouzi, B.; Mirzakuchaki, S.; Seyedzadeh, S.M. & Mosavi, M.R. A simple, sensitive and secure image encryption algorithm based on hyper-chaotic system with only one round diffusion process. *Multimedia Tools App*., 2014, **71**(3), 1469–1497. doi:10.1007/s11042-012-1292-9
12. Liu, H.; Kadir, A. & Liu, J. Color pathological image encryption algorithm using arithmetic over Galois field and coupled hyperchaotic system. *Opt. Lasers Eng*., 2019. **122**, 123–133. doi:10.1016/j.optlaseng.2019.05.027
13. Rukhin, A.; Soto, J.; Nechvatal, J.; Smid, M. & Barker, E. A statistical test suite for random and pseudorandom number generators for cryptographic applications, NIST Special Publications, Va, USA, 2010. 71p.
14. Dougherty, S.T.; Klobusicky, J. & Şahinkaya, S. An S-Box construction from exponentiation in finite fields and its application in RGB colour image encryption. *Multimedia Tools App*., 2023, **83**(14), 1-29. doi:10.1007/s11042-023-17046-6
15. Lavanya, M.; Joseph, Abraham.; Sundar, K. & Saravanan, S. Simplified Image Encryption Algorithm (SIEA) to enhance image security in cloud storage, *J. Multim. Tools and Appl.*, 2024, 1573-7721. doi:10.1007/s11042-023-17969-0
16. Adeniyi, A.E.; Abiodun, K.M. & Awotunde, J.B. Implementation of a block cipher algorithm for medical information security on cloud environment: Using modified advanced encryption standard approach. *Multimed Tools Appl*., 2023, **82**(10), 1-5. doi:10.1007/s11042-023-14338-9
17. Anushiadevi, R. & Amirtharajan, R. Design and development of reversible data hiding-homomorphic encryption & rhombus pattern prediction approach. *Multimed Tools Appl*., 2023, **82,** 46269–46292. doi:10.1007/s11042-023-15455-1
18. Ratan, R. & Yadav, A. Security analysis of bit plane level image encryption schemes. *Def. Scie. J*., 2021, **71**(2), 209-221. doi: 10.14429/dsj.71.15643
19. Teng, L.; Wang, X. & Xian, Y, Image encryption algorithm based on a 2D-CLSS hyperchaotic map using simultaneous permutation and diffusion. *Inf. Sci*., 2022, **605**, 71–85. doi:10.1016/j.ins.2022.05.032
20. Wei, Song.; Chong, Fu.; Ming, Tie.; Chiu-Wing, Sham. & Jun, Liu. A fast parallel batch image encryption algorithm using intrinsic properties of chaos. *Signal Process. Image Commun*., 2022, **102**, 116628. doi:10.1016/j.image.2021.116628

CONTRIBUTORS

Dr M. Lavanya is presently working as an Assistant Professor at School of Computing, SASTRA Deemed University, Thanjavur, Tamil Nadu. Her research interest focuses in Cryptography, cloud resource allocation, cloud security and operating system principles.
In this paper she has contributed in development of the encryption and decryption algorithm. She has been the content writer of this paper.

Dr K. Joseph Abraham Sundar is currently working as an Assistant Professor at the School of Computing, SASTRA Deemed University, Thanjavur, Tamil Nadu. He obtained his PhD in Computer Science Engineering from SASTRA Deemed University. His areas of interests are super resolution image reconstruction, text and object detection, image cryptography. In the current study he has carried out the performance analysis of the encryption and decryption algorithms. He has contributed significantly in the image processing aspects of the paper.

Dr S. Saravanan is presently working as an Assistant Professor at Department of Electronics and Communication Engineering, Srinivasa Ramanujan Centre, SASTRA Deemed University, Kumbakonam, Tamil Nadu.
In the current study he has done extensive literature survey in order to identify the merits and demerits of various encryption methods which has been a great support in this research work.