

A Secure Image Encryption Algorithm Based on Hill Cipher System

S.K.Muttoo¹, Deepika Aggarwal², Bhavya Ahuja³

^{1,2,3}Department of Computer Science, University of Delhi, India

e-mail: skmuttoo@cs.du.ac.in¹, success4deepika@yahoo.co.in², success_bhavya19@yahoo.com³

Abstract

We present a technique of image encryption based on Hill cipher system that provides better security than existing approach of Bibhudendra Acharya et al. by rendering the image content completely scrambled using multiple self-invertible keys, block shuffling and a new developed pel transformation. The Hill cipher algorithm is one of the symmetric key algorithms having several advantages in encryption. However, the inverse of the matrix used for encrypting the plain text in this algorithm may not always exist. Moreover this algorithm is susceptible to known plain text attack. Our proposed algorithm is aimed at better encryption of all types of images even ones with uniform background and makes the image encryption scheme more secure.

Keywords: Cryptography, Hill Cipher, Image Encryption, pel transformation

1. Introduction

With the rapid advancement in network technology especially Internet, it has become possible to transmit any type of data across networks. This has raised concern for the security of the transmitted data as access to data which has become easier by interception of communication media. Hence, data security is becoming an imperative and critical issue in data storage and transmission to prevent it from attacks. Images are widely used in several processes. Therefore, the protection of image data from unauthorized access is important.

Encryption refers to the algorithmic schemes that encode the original message referred to as plain text using a key into non-readable form, a coded message known as cipher text so that it is computationally infeasible to be interpreted by any eavesdropper. The receiver of the cipher text uses a key to retrieve back the message in original plain text form [6][7].

Substitution cipher is one of the basic components of classical ciphers. A substitution cipher is a method of encryption by which units of plain text are substituted with cipher text according to a regular system; the units may be single letters (the most common), pairs of letters, triplets of letters, mixtures of the above, and so forth. The receiver decipheres the text by performing an inverse substitution. The units of the plain text are retained in the same sequence as in the cipher text, but the units themselves are altered.

For substituting large group of letters, we use polygraphic substitution ciphers. We also have mono-alphabetic substitution ciphers that use a fixed substitution over the entire message. Hill cipher is one of such ciphers and we have used this for image encryption in the paper.

The Hill cipher has several advantages such as disguising letter frequencies of the plain text, using simple matrix multiplication and inversion for enciphering or deciphering, high speed and high throughput. But some problems have been noticeable in the encryption scheme. Inverse of a matrix may not exist due to which decryption will not be possible. Due to its linear nature, it succumbs to known-plain text attack. On application of the encryption algorithm which images with uniform background, the images could not be encrypted properly as pixels with similar intensity values (as with uniform background) map against to similar intensity values. In order to address these issues and enhance secrecy of encrypted data using Hill cipher, we are proposing an algorithm which uses a different Self-Invertible Matrix Generation Method for Hill cipher system. This method can be used multiple times to generate a different self-invertible matrix for each block of the image. We have also applied a new pel transformation and block

shuffling in the algorithm. Our algorithm works well for all types of gray scale as well as color images [1][2].

The organization of the paper is as follows. The present section i.e. Section 1 is the introductory. A brief review of Hill cipher is given in Section 2. The proposed method for encrypting and decrypting the images has been discussed in Section 3. Section 4 summarises the experimental results of proposed algorithm.

2. Hill Cipher

The Hill cipher is a polygraphic block cipher based on linear algebra developed by Lester Hill [1] in 1929. Using frequency analysis, substitution ciphers like mono-alphabetic ciphers can be easily broken. But Hill cipher completely hides single letter frequencies by encrypting pairs of plain text and so it's safe against cipher-text only attacks. It provides good diffusion as change in one letter of plain text affects all letters in the cipher text. All arithmetic is done modulo some integer z that is the total number of possible symbols.

For encryption, the algorithm takes m successive plain text letters and instead of that substitutes m cipher letters. Each character is assigned a numerical value like $a = 0$, $b = 1$ and so on. The substitution of cipher text letters in the place of plain text letters leads to m linear equation. For $m = 3$, the system can be described as follows:

$$\begin{aligned} C_1 &= (K_{11}P_1 + K_{12}P_2 + K_{13}P_3) \text{ mod } 26 \\ C_2 &= (K_{21}P_1 + K_{22}P_2 + K_{23}P_3) \text{ mod } 26 \\ C_3 &= (K_{31}P_1 + K_{32}P_2 + K_{33}P_3) \text{ mod } 26 \end{aligned}$$

This case can be expressed in terms of column vectors and matrices:

$$\begin{pmatrix} C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{bmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{bmatrix} \begin{pmatrix} P_1 \\ P_2 \\ P_3 \end{pmatrix}$$

or simply $C = KP$, where C and P are column vectors of length 3, representing the plain text and cipher text respectively, and K is a 3×3 matrix, which is the encryption key. All operations are performed mod 26 here. Decryption requires using the inverse of the matrix K . The inverse matrix K^{-1} of a matrix K is defined by the equation $KK^{-1} = K^{-1}K = I$, where I is the Identity matrix. But the inverse of the matrix does not always exist, and when it does, it satisfies the preceding equation. K^{-1} is applied to the cipher text, and then the plain text is recovered [4] [5]. In general term, we can write as follows:

For encryption: $C = E_k(P) = KP$

For decryption: $P = D_k(C) = K^{-1}C = K^{-1}KP = P$

If the block length is m , there are 26^m different m letters blocks possible, each of them can be regarded as a letter in a 26^m letter alphabet. In this paper, we have used Hill Cipher for encryption of images. All arithmetic has been done modulo 256 (8 bits per pixel for gray scale and 8 bits for RGB per color component).

3. Proposed Secure Image Encryption Algorithm

In this section, we propose an encryption scheme based on Hill Cipher involving multiple key generation, pel transformation and block shuffling. The scheme intends to address some issues in the cipher scheme using a single self invertible key matrix for encryption in [3].

The scheme proposed in [3] was susceptible to Known Plain Text attack in which the key can be found by attacker using some known plain text-cipher text pairs. Also the images with uniform background could not be encrypted properly. The images with very close pixel values or equal values map to similar values or to values that have very low perceptual difference giving very poor encryption results as is demonstrated in Figure 1. The pseudocode for the proposed encryption algorithm is given in Figure 2.

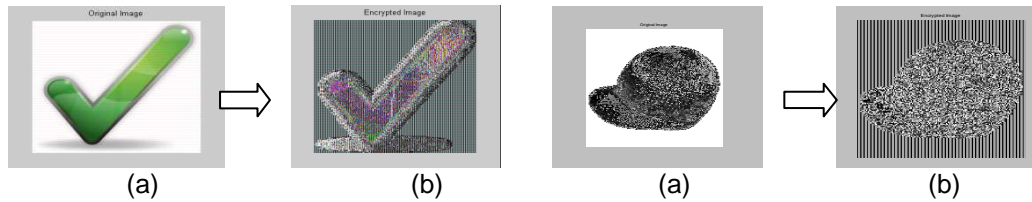


Figure 1. Encryption Result: (a) Original Image, (b) Encrypted Image

```

Input : The image to be encrypted
Output : The Encrypted image
Begin
Divide the image into 8X8 blocks
Do for each block
Generate a 4X4 self -invertible key K for the block using the algorithm given in [3]
Take four adjacent pixel values and if within epsilon threshold apply pel transformation
Encrypt the pixel values using the key K as C = K*values mod 256
Endo
Create a new image with these new pixel values and shuffle 8X8 blocks of the new image
using a shuffling key
End
    
```

Figure 2. Pseudocode for proposed image encryption algorithm

The algorithm uses the multiple self-invertible key generation method as proposed in [3] to generate a different key for each block.

3.1 Pel Transformation

If the pixel values are identical or very close, then after encryption they map to very close values or even to the same values as before encryption in case of equal intensity values. This is noticeable from the following example. Using constant k=1 in self-invertible key generation method:

Say Key =
$$\begin{pmatrix} 12 & 120 & 245 & 136 \\ 176 & 224 & 80 & 33 \\ 13 & 120 & 244 & 136 \\ 176 & 225 & 80 & 32 \end{pmatrix}$$

and Pixel Values = [20 20 20 20]. Then, $NewPixelValues^T = Key * Values^T \text{ mod } 256 = [20 \ 20 \ 20 \ 20]$

This results in poor encryption results as old and new pixel values are same. So we propose that application of some pixel value transformations before encryption would result in better results. Let epsilon be the difference between values of adjacent pixels in a quadruple of pixel values to be encrypted. If all the differences between the four adjacent pixel values is less than epsilon, then pixel value transformation function T given in (1) below is applied.

$$v_i' = T(v_i) = \begin{cases} v_i * z^{i-1} \text{ mod } 256 & \text{if indicator} = 1 \\ v_i & \text{otherwise} \end{cases} \quad \text{for } i=1,2,3,4 \quad (1)$$

$$indicator = \begin{cases} 1 & \text{if } abs(v_i - v_{i-1}) \leq \epsilon \text{ for } i=2,3,4 \\ 0 & \text{otherwise} \end{cases}$$

Here z is a random number except 0 and 1. z values for all blocks are also transmitted in an array multiplier. With this transformation the perceptual difference between the pixel values is increased and they are mapped to uncorrelated values in the domain $[0,255]$. Hence, after encryption they result in varied uncorrelated intensities. We illustrate this with an example.

Example 3.1

$$\text{Let Key} = \begin{pmatrix} 5 & 35 & 252 & 221 \\ 245 & 179 & 11 & 78 \\ 6 & 35 & 251 & 221 \\ 245 & 180 & 11 & 77 \end{pmatrix}$$

It can be seen that the pixel values $v = [181 \ 185 \ 183 \ 180]$ are encrypted as $\text{new}v = [118 \ 107 \ 36 \ 78]$ when $z=7$ and $v' = [181 \ 15 \ 7 \ 44]$. If transformation T had not been applied, v would have been encrypted as $\text{new}v = [92 \ 73 \ 90 \ 78]$. Also the transformation increases the secrecy of the encryption scheme as the pixel values encrypted may not be equal to the original image pixel values.

3.2 Shuffling Image Blocks

A random permutation on the number of blocks is generated and the new block values are stored in array pos which serve as the shuffling key. The blocks then shuffled according to these new block positions. These results, in more secrecy as the blocks, are randomly distributed throughout the image. Figure 3 demonstrates the block diagram for the proposed encryption algorithm.

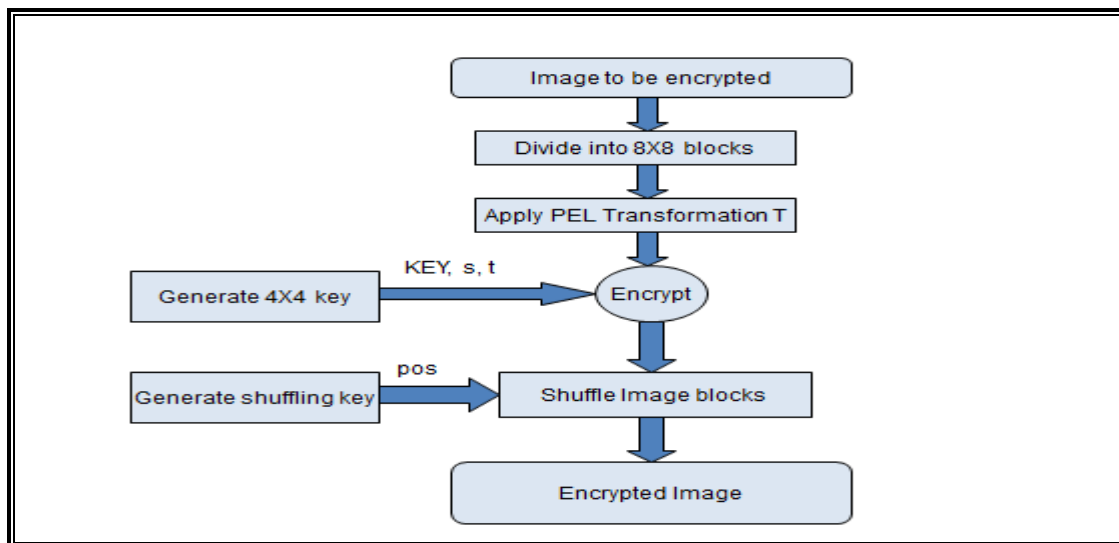


Figure 3. Block diagram for encryption process of Proposed Algorithm

The sender transmits the encrypted image through the communication channel to the receiver. The block seeds and t values used for key generation, pos and multiplier are transmitted through a secure channel.

For RGB images the three color components are separately encrypted and the three components form the new RGB value.

3.3. Decryption

The pseudocode for the proposed hill cipher decryption algorithm has been given in Figure 4.

Input: The image to be decrypted, block seed and t values, pos, multiplier

Output: The Original image

Begin

Reshuffle the image blocks to their original locations using shuffling key pos

Divide the image into 8X8 blocks

Do for each block

Generate the 4X4 self-invertible key for the block using seed and the using of the key generation algorithm given in [3]

Decrypt four pixel values at a time from the block

Approximate the new pixel values if value of multiplier is not 1

Endo

Create the new decrypted image with these new pixel values

End

Figure 4. Pseudocode for image decryption using proposed variant of hill cipher

After reshuffling to original locations using the shuffling key pos, the blocks are relocated to their original positions. The image is divided into 8X8 blocks and then four pixel values are taken at a time. The original key is generated using the received block seed and t values is used to compute $K * \text{values mod } 256$ to get back the encrypted values.

The value of multiplier denotes the value of z used for transformation T outlined in (1). If the value is not equal to one then pel transformation has been applied. For these pixels we only get the first decrypted value equal to the original value. It is not possible to obtain correctly the original value for the other three pixels due to the mod operation.

Hence, we approximate the other values to the first value taking into account the limitations of Human Visual System as differences of the order of epsilon are non-perceptible. The decryption results will hence depend on the value of epsilon. Block diagram for decryption process is given in Figure 5.

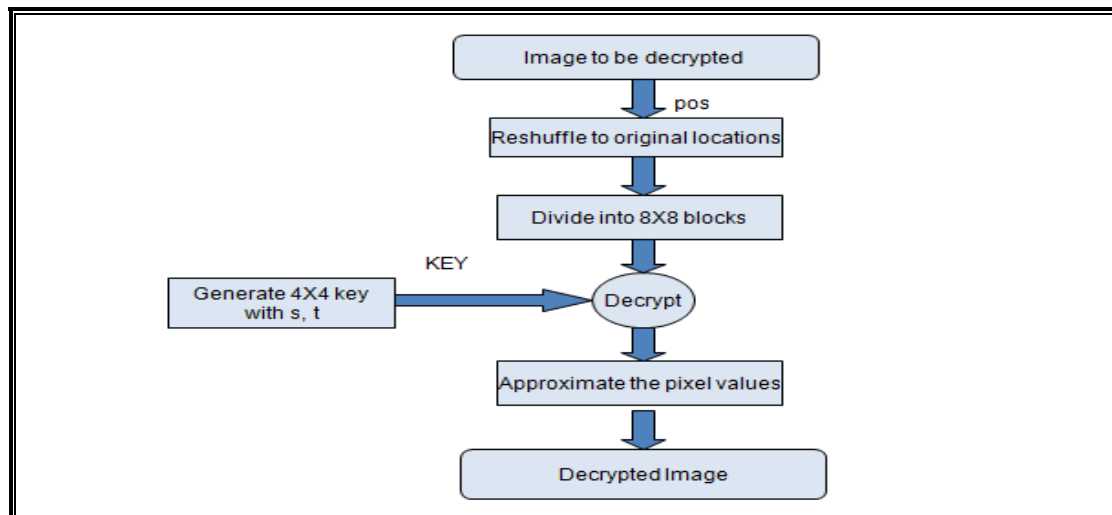


Figure 5. Block diagram for decryption process of proposed algorithm

4. Experimental Results

This section represents the simulation results illustrating the performance of the proposed encryption algorithm. The encryption and decryption algorithms are implemented in MATLAB 7.7.0(R2008b). The encryption and decryption results using the proposed algorithm are given in Figure 6.

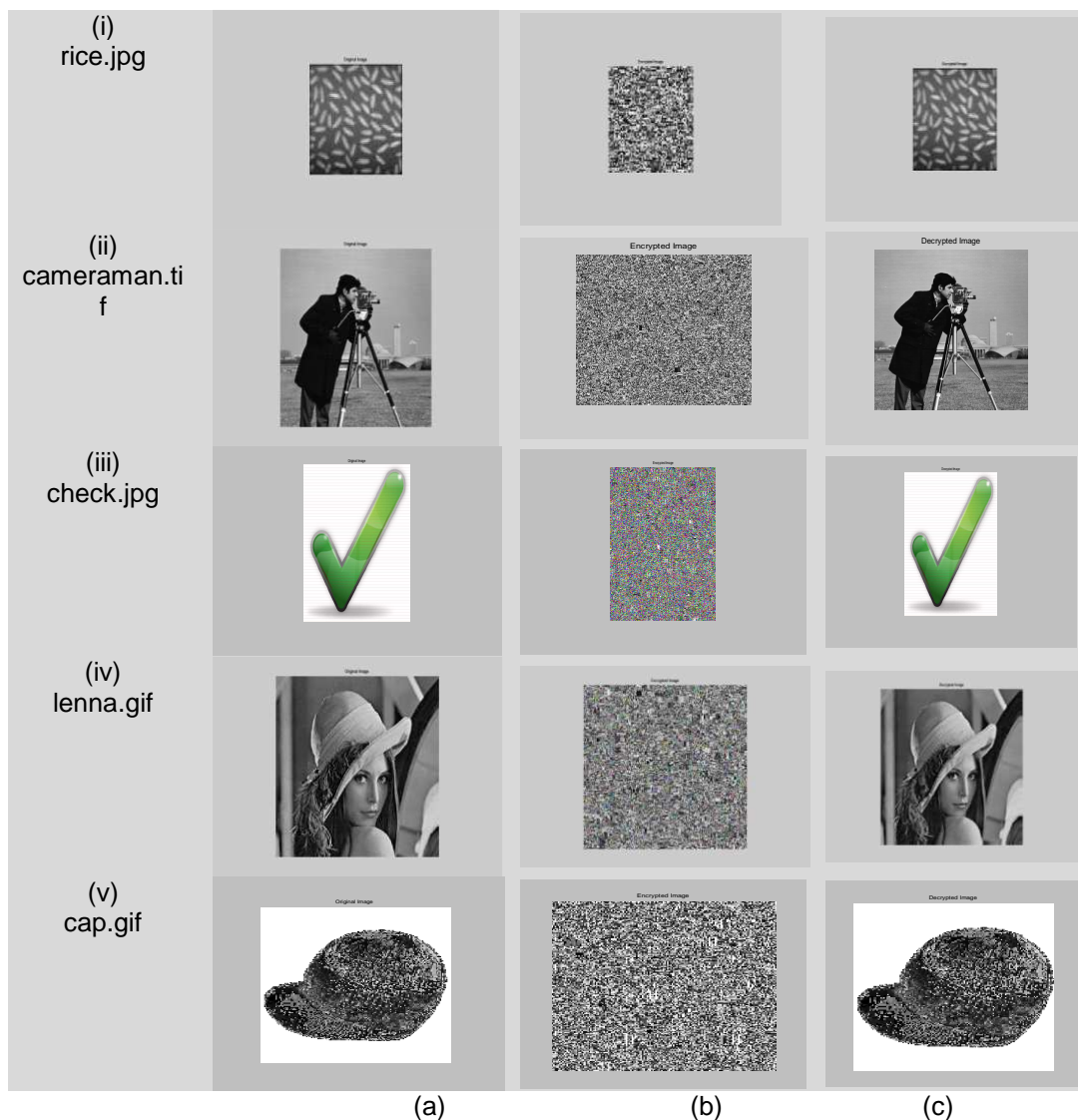


Figure 6. Encryption and decryption results using proposed algorithm: (a) Original Image, (b) Encrypted Image, (c) Decrypted Image

The more features of an image are hidden better than cryptosystem. But mere visual inspection is not a good performance evaluator. A good encryption scheme must be robust against all kinds of cryptanalytic, brute force and statistical attacks. Here, we analyzed the performance of the proposed encryption scheme.

4.1 Histogram analysis

The histogram analysis clarifies how pixels in an image are distributed by plotting the number of pixels at each intensity level. Histogram analysis on test images check.jpg and cap.gif respectively using plain hill cipher algorithm and proposed algorithm is given in Figure. 7 and 8 respectively. The histogram of encrypted image has uniform distribution which is significantly different from original image and has no statistical similarity in appearance. Relatively uniform distribution in encrypted image histogram using proposed algorithm as compared to plain hill cipher algorithm points out good quality of encryption method.

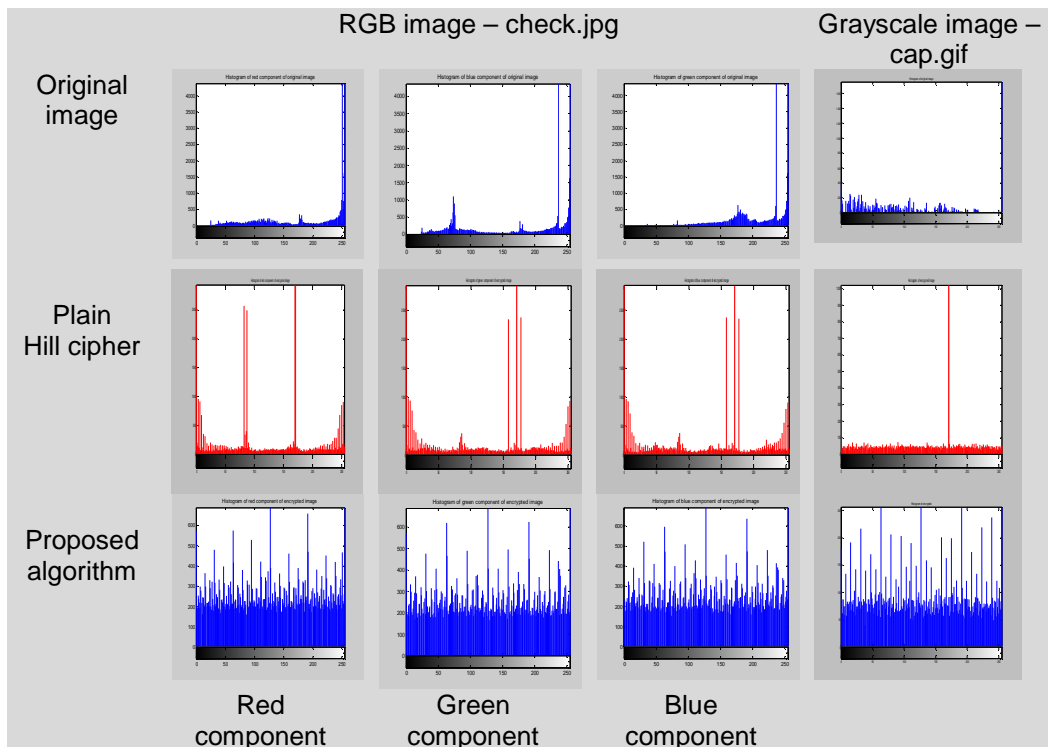


Figure 7. Histogram results on application of plain Hill cipher and proposed algorithm

4.2 MSE and PSNR

Mean Square Error (MSE) is the cumulative squared error between original image f and decrypted image f' . A lower value of MSE means less error in decryption of the image. The formula used is given below:

$$MSE = \frac{1}{N \cdot M} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} [|f(i, j) - f'(i, j)|^2]$$

The image is size $M \times N$. For RGB images mean of the three MSE values for the R, G and B components is taken to be the final MSE. Peak Signal to Noise Ratio can be used as a measure of recovered image quality. It is calculated using MSE as:

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right) \\ = 20 \cdot \log_{10} \left(\frac{MAX_I}{\sqrt{MSE}} \right)$$

Here, MAX_I is the maximum possible pixel value of the image. When the pixels are represented using 8 bits per sample, this is 255. For color images the definition of PSNR is the same, except the MSE is the sum over all squared value differences divided by image size and by three. Figure 9. shows the graph exhibiting the relationship between epsilon and MSE (shown in blue color), PSNR (shown in green color) values for some grayscale and color images of different resolution. It can be seen that the mean square error in decryption is very low for small values of epsilon and the quality of the decrypted image is sufficiently good as shown by the PSNR values. By increasing epsilon, it increases the MSE. This is attributed to the fact that for a greater value of epsilon, the pel transformation is applied on a greater number of pixel values. Due to this, the next three pixel values of the quad being encrypted together are set to be first pixel value of the quad. This results in an increase in the difference between the actual pixel value and the value decrypted which leads to an increase in MSE.

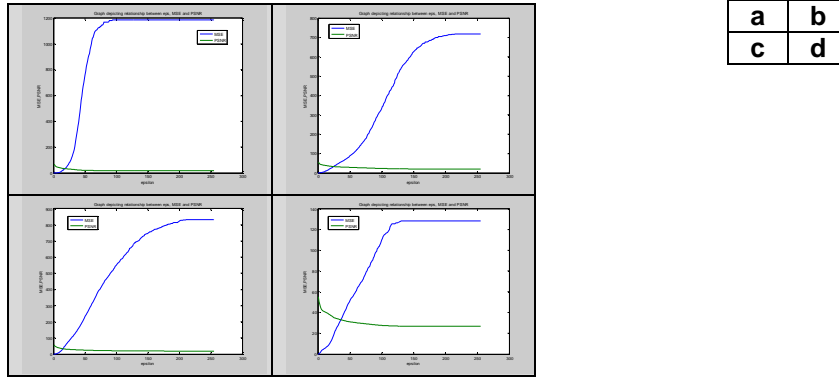


Figure 9. Graph depicting relationship between epsilon and MSE, PSNR for images, (a) rice.jpg, (b) cameraman.tif, (c) lenna.gif, (d) check.jpg

4.3 Correlation

A good encryption algorithm must generate an encrypted image independent of the original image. So they must have a very low correlation coefficient which is very close to zero. Here, we have calculated the correlation between original and encrypted image. A plot showing values of correlation coefficient on varying epsilon for some grayscale and color images as has been given in Figure 10. A low value of correlation coefficient shows that there is no straight relation between the original and encrypted images. The formula used to calculate correlation coefficient is given as:

$$C.C = \frac{\sum_{i=1}^N (x_i - E(x))(y_i - E(y))}{\sqrt{\sum_{i=1}^N (x_i - E(x))^2} \sqrt{\sum_{i=1}^N (y_i - E(y))^2}}$$

C.C: correlation coefficient

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

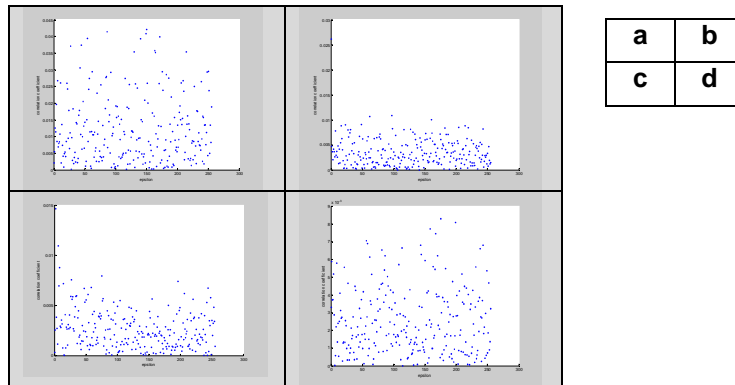


Figure 10. Plot showing values of correlation coefficient on varying epsilon for images, (a) rice.jpg, (b) cameraman.tif, (c) lenna.gif, (d) check.jpg

In Table I, we give the comparison of the correlation coefficient between original image and encrypted image using the plain Hill cipher algorithm and proposed crypto-algorithm. A lower value of the correlation coefficient obtained in case of our algorithm as compared to that obtained in the plain Hill cipher algorithm indicates that the encrypted image is less correlated to the original image resulting in better encryption.











4.4 Entropy

Entropy is a statistical measure of randomness that can be used to characterize the texture of an image. The entropy of an image A is defined as:

$$H_e = - \sum_{k=0}^{G-1} P(k) \log_2 (P(k))$$

where H_e is the entropy, G is gray value of input image (0...255), $P(k)$ is probability of the occurrence of symbol k . Higher value of entropy of encrypted image, better the security. In Table I, we give the comparison of the entropy between original image and encrypted image using the plain Hill cipher algorithm and proposed crypto-algorithm. A higher value of the entropy obtained in case of our algorithm as compared to that obtained in the plain Hill cipher algorithm indicates that our algorithm introduces more randomness in the encrypted image resulting in better encryption.

Table I. Entropy and Correlation coefficient between original image and encrypted image using plain hill cipher and proposed algorithm

IMAGES	ENTROPY		CORRELATION COEFFICIENT BETWEEN ORIGINAL AND ENCRYPTED IMAGE		
	Original image	Encrypted image		Plain Hill cipher	Proposed algorithm (eps=10)
		Plain Hill cipher	Proposed algorithm		
 check.jpg	4.8936	5.5694	7.9172	-0.0328	0.0034
 cap.gif	4.5303	6	8	-0.0596	-3.3410e-005
 donald.jpg	3.0886	4.7041	7.8003	-0.0499	-0.0055
 pout.tif	5.7599	8	8	-0.0425	0.0025
 cycle.tif	3.6526	5.0706	7.8377	-0.0493	-0.0035
 insect.bmp	6.7786	7.4476	7.8449	0.0554	-0.0114
 rose.gif	2.9579	5.0330	7.7599	-0.0526	0.0141
 thmb.png	5.8024	6.4375	7.9226	-0.0276	-0.0016
 twitter.jpg	2.4010	3.7810	7.7919	-0.2896	9.9401e-004
 flower.png	7.3180	7.6252	7.9067	0.0656	-5.9256e-004

4.5 Structural Similarity Index Measure (SSIM)

SSIM is a method for measuring the similarity between two images. It is a full reference metric. SSIM is designed to improve on traditional methods like PSNR and MSE which have proved to be inconsistent with human eye perception. The SSIM metric is calculated on various windows of an image. The measure between two windows x and y of common size $N \times N$ is:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

with μ_x the average of x ; μ_y the average of y ; σ_x^2 the variance of x ; σ_y^2 the variance of y ; σ_{xy} the covariance of x and y ; $c_1=(k_1L)^2$, $c_2=(k_2L)^2$ two variables to stabilize the division with

weak denominator; L the range of the pixel-values (typically this is $2^{\#bits \text{ per pixel}} - 1$); $k_1 = 0.01$ and $k_2 = 0.03$ by default.

The resultant SSIM index is a decimal value between -1 and 1, and value 1 is only reachable in the case of two identical sets of data. We have calculated the SSIM between original and decrypted images. It can be seen that the values are very close to one for low values of epsilon. Figure 11 shows the graph depicting relationship between epsilon and SSIM. It can be seen that on increasing epsilon the value of SSIM decreases.

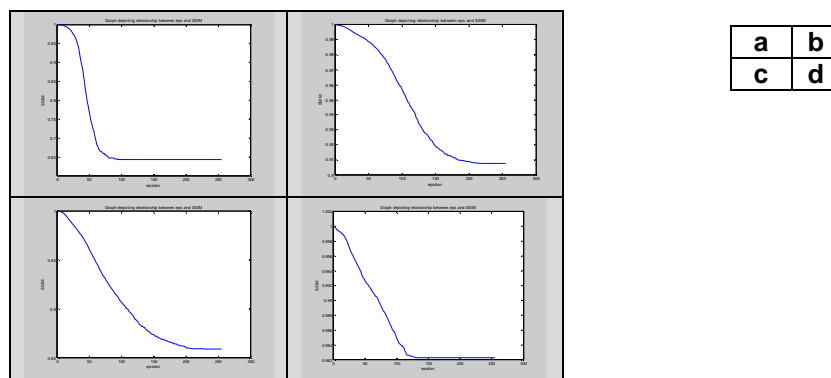


Figure 11. Graph depicting relationship between epsilon and SSIM: (a) rice.jpg, (b) cameraman.tif, (c) lenna.gif, (d) check.jpg

5. Conclusion

In this paper, a crypto-algorithm based on hill cipher is presented. The proposed cryptosystem uses a different key for each block encryption and the possibility of known plain text attack is highly reduced as the key used changes with every block and it is generated randomly using a seed and multiplier. Also in some cases, the encrypted pixel values are not the original intensity values but obtained from pel transformations. With block shuffling, the algorithm becomes more secure. For an image with uniform background, the results are improved due to changing keys, block shuffling and applying new developed pel transformation. The perceptual difference between close pixel values increases on applying pel transformation.

There may be cases in which the decrypted results are not completely similar to the original image. This is because the decrypted values are not exactly equal to the original values, in case the four values are within epsilon threshold, so in some regions like edges, degradations from the original image are noticeable. With a smaller value of epsilon the decryption results will improve. Hence, there is a trade-off between better encryption and better decryption results.

References

- [1] Bibhudendra Acharya, Saroj Kumar Panigrahy and Debasish Jena. Image encryption using self invertible key matrix of Hill cipher algorithm. *1st International Conference on Advances in Computing*. Chikhli, India. 21-22 February 2008.
- [2] Bibhudendra Acharya, S K Patra, G. Panda. A Novel cryptosystem using matrix self invertible key matrix of Hill cipher algorithm. *1st International Conference on Advances in Computing*, Chikhli, India. 21-22 February 2008.
- [3] Lerma M A. Modular Arithmetic. 2005. Available on: http://www.math.northwestern.edu/~mlerma/problem_solving/results/modular_arith.pdf
- [4] Petersen K. Notes on Number Theory and Cryptography. 2000. Available on: <http://www.math.unc.edu/Faculty/petersen/Coding/cr2.pdf>.
- [5] Eisenberg M. Hill Ciphers and Modular Linear Algebra. 1999. Available on: <http://www.apprendre-en-ligne.net/crypto/hill/Hillciph.pdf>.
- [6] Lester S Hill. Cryptography in an algebraic alphabet. *Amer. Math.* 1929; 36: 306-312.
- [7] William Stallings. Cryptography and Network Security. Fourth Edition. Prentice Hall. 2005.