Two Level Image Encryption using Pseudo Random Number Generators

Vishal Kapur
Dept. of CSE
Manipal Institute of Technology
Manipal University, Manipal

Surya Teja Paladi
Dept. of CSE
Manipal Institute of Technology
Manipal University, Manipal

Navya Dubbakula
Dept. of CSE
Manipal Institute of Technology
Manipal University, Manipal

ABSTRACT

The escalating significance of securing data over the network has encouraged development of secure encryption algorithms. Cryptography is a technique that assists in development of such secure algorithms. It involves converting intelligible data into an unintelligible form to protect it from intruders. In this paper, the authors propose a simple and secure process to secure images. The image encryption process makes use of two Pseudo Random Number generators. In the first step, Linear Feedback Shift Register algorithm is used to swap the rows of the original image. This is followed by the swapping of the columns to produce an intermediary cipher image. In the second step, Blum Blum Shub algorithm is used to substitute the intensity of each pixel of the intermediary cipher image, to produce the final encrypted image. Various analysis tests are then performed to check the quality of the encrypted image. The test results prove the efficiency of the proposed image encryption process.

Keywords

Cryptography, Pseudo random number generator, Linear Feedback Shift Register, Blum Blum Shub, Image encryption, Image Decryption

1. INTRODUCTION

Activity of exchanging information and thoughts over space is known as communication. Secure communication involves protecting this exchange of information from intruders. Some mechanism needs to be provided in order to achieve this security. Cryptography provides this much needed data security [1].

Cryptography is a technique that involves converting intelligible data such as text, pictures, audio or video into an unintelligible form in order to protect it from attackers. It is probably the most important means to provide secure communication. The intelligible data in a cryptosystem is called plaintext and its encrypted unintelligible form is known as ciphertext. Any encryption algorithm takes plaintext and a secret key as input and produces the ciphertext as output. The secret key is known only to the sender and the receiver.

There are two methods to achieve encryption - substitution and transposition [2]. Substitution involves mapping the elements of plaintext into other elements whereas transposition involves swapping the elements of the plaintext.

Image Encryption is the process of encoding the original image in such a way that only authorized people can read it. It involves applying various substitution or transposition methods to convert the plain image into a cipher image. Image Encryption is extensively used in various military applications and for secure image transmission over the internet.

This paper makes use of a symmetric encryption [3] process in which the same key is used for encryption as well as decryption. It uses two pseudo random number generators (PRNG) to achieve encryption [4].PRNG is a process of generating a set of numbers that possess two important properties – randomness and unpredictability [5]. This means that an attacker cannot guess the next number in the sequence even with the knowledge of some of the numbers of that sequence.

The two Pseudo Random Number Generators used in this paper are Linear Feedback Shift Register [6] [7] and Blum Blum Shub [8].

The rest of the paper is divided into the following sections. Section 2 highlights some of the related work done in this area. Section 3 describes the proposed image encryption algorithm in detail. Section 4 contains the resulting images obtained at every step of encryption and decryption. Section 5 presents the experimental results and security analysis. Finally, section 6 concludes the work.

2. RELATED WORK

Authors in [9] suggested two techniques to encrypt images and both the techniques used an image as a key. Each pixel of the original image was XORed with each pixel of the key image to obtain the encrypted image.

Authors in [10] have suggested the use of a cover image to encrypt another image. The cover image will be used for hiding the original image. The only drawback in the proposed technique is that the cover image and original image should be of the same size.

Authors in [11] have proposed an image encryption algorithm based on three different chaotic maps. The original image is first divided into 8X8 size blocks. Next, block shuffling is introduced using 2D Cat map .The shuffled image is encrypted using chaotic sequence generated by 1D logistic map.

In this paper, we propose an image encryption process that uses both substitution as well as transposition which in turn provides better security than any one of them being used.

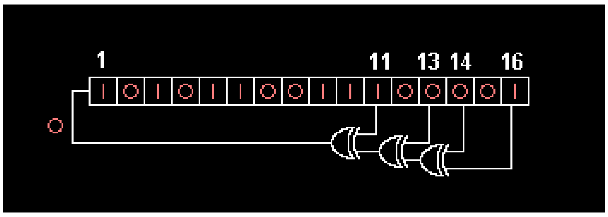
3. PROPOSED WORK

3.1 Encryption

A two level image encryption algorithm is used to convert an image into its distorted form.

3.1.1 Linear Feedback Shift Register

Linear feedback shift register (LFSR) algorithm is used for the first level of encryption. This algorithm takes a seed as an input. This seed forms the first number of the sequence generated. LFSR makes us of a linear feedback function (ex : $x^{16} + x^{14} + x^{13} + x^{11} + 1$) which represents the feedback bits which are to be XORed. 1 (x^0) in the polynomial represents the left most bit, where the value after the XOR operation is placed. The bits are then right sifted by 1 position and the XOR value is kept as the first bit. This becomes the next number in the sequence.



Linear feedback function: $x^{16} + x^{14} + x^{13} + x^{11} + 1$
Current state: 1010110011100001
Next state: 0101011001110000

As the powers in the linear feedback function are 16,14,13,11, the bits at these positions are XORed. Now the bits are right shifted by 1 and the XORed value is kept as the first bit.

ALGORITHM: Linear Feedback Shift Register Algorithm
Input: seed
Output: sequence of pseudo random numbers
Initialize: **N**um ← **s**eed

```
1. While Num !=0 and Num is not repeating do
2.     bin ← Obtain the binary pattern of the Num
3.     Pad with leading zeros until bin has 16 bits.
4.     XOR the bits at positions corresponding to linear
       feedback function and store it in m.
5.     bin ← Right shift the bin by 1.
6.     Pad with leading zeros until bin has 16 bits.
7.     Replace the first bit with m in bin.
8.     Num ← Obtain the decimal value of the binary pattern
       obtained.
9. End
```

3.1.2 First Level of Encryption

In the first level of encryption, the positions of the pixels are changed by permuting the rows and columns of the original image. This is done by making use of the random numbers generated by LFSR. First the rows are permuted followed by the columns. During row permutation the numbers in the sequence are restricted to the height of the image by using the modulus function. Then each row is swapped with the row having row number equal to the number generated after the modulus operation. This process continues until all the rows are swapped. This step of encryption yields a row distorted image.

ALGORITHM: Row Permutation Algorithm
Input: array generated from LFSR, Image
Output: Row distorted image
Initialize: k←0

```
1. While k< (height of image)
2.     Obtain pixel value of the each pixel in k^th row.
3.     Obtain pixel value of each pixel in the row number
       equal to number generated after modulus function at that
       instant.
4.     Swap the pixel values generated in step2 and step3.
5.     Increment k.
6. End
```

The next step involves the same procedure of permuting the columns but instead of height, the width of the original image is considered. Each column is swapped with another column whose column number equals number generated after the modulus operation. The image obtained after this step of encryption is the intermediate cipher image.

ALGORITHM: Column Permutation Algorithm
Input: array generated from LFSR, Row distorted Image
Output: Intermediate image
Initialize: k←0

```
1. While k< (width of image)
2.     Obtain pixel value of the each pixel in k^th column.
3.     Obtain pixel value of each pixel in the column number
       equal to number generated after modulus function at
       that instant.
4.     Swap the pixel values generated in step2 and step3.
5.     Increment k.
6. End
```

3.1.3 Blum Blum Shub

Blum Blum Shub algorithm is used for the second level of encryption. This algorithm was proposed by Lenore Blum, Manuel Blum and Michael Shub in 1986.

Let p and q be two large prime numbers having reminder 3 when divided by 4.

$$p \equiv q \equiv 3 \pmod 4$$

Then the sequence is generated by the following equation:-

$$X_{n+1} = X_n^2 \bmod M$$

Where M is the product of p and q .Initial seed X_0 is randomly chosen in such a way that it is relatively prime to M which means that neither p nor q are a factor of s

ALGORITHM: Blum Blum Shub Algorithm
Input: p, q, X_0
Output: sequence of pseudo random numbers
Initialize: M← p x q
Num ← (X_0 * X_0) mod M

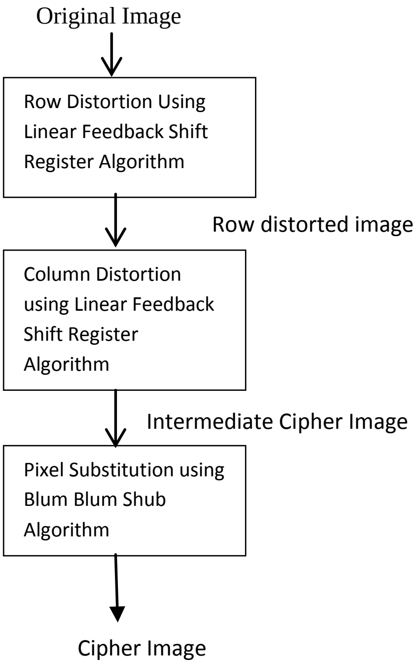
```
1. While Num! =0 and Num is not repeating do
2.     Store the value of Num
3.     Num ← (Num*Num) mod M
4. End
```

3.1.4 Second Level of Encryption

The authors use random numbers generated by the BBS algorithm for the second level of encryption. In this level of encryption, xor operation is used to substitute the intensity of each individual pixel of the intermediary cipher image obtained. The numbers obtained from the BBS exceed the pixel intensity range (0 to 255), so the numbers are restricted within the range of 0 to 255 using the modulus operation. Next the intensity of each pixel is XORed with the number obtained after the modulus operation. This new number obtained is set as the intensity of that pixel in the image. For each pixel the array index is incremented and this procedure is carried on until all the pixels are substituted.

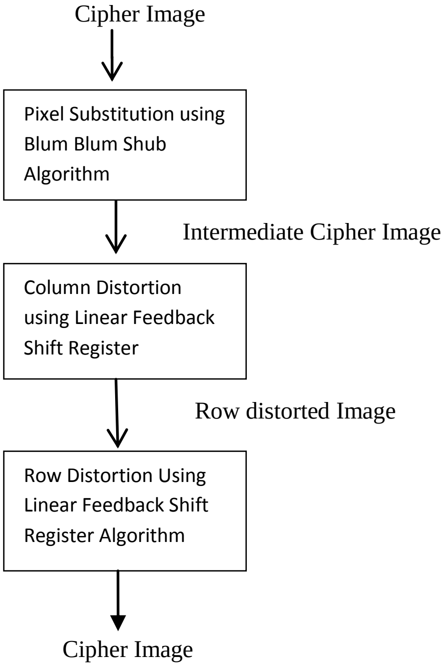
ALGORITHM: Pixel Substitution Algorithm
Input : array generated from BBS, Intermediate Cipher Image
Output : Cipher Image
Initialize : k←0

```
1. While k< (width*height of the image)
2.     Obtain intensity value of the k^th pixel.
3.     Restrict the number obtained at that instant to range 0
       to 255 using modulus function
4.     XOR the numbers obtained in step2 and step3.
5.     Substitute the pixel with intensity value obtained in
       step4
6.     Increment k.
7. End
```


Original Image
Row Distortion Using Linear Feedback Shift Register Algorithm
Row distorted image
Column Distortion using Linear Feedback Shift Register Algorithm
Intermediate Cipher Image
Pixel Substitution using Blum Blum Shub Algorithm
Cipher Image

3.2 Decryption

The decryption procedure involves applying the algorithms used in the reverse order, that is, first BBS algorithm is used followed by LFSR algorithm. Same keys are used for the decryption process with the only difference being that the random numbers generated by the algorithms are used in the reverse order as compared to the encryption process. Decryption procedure results in the reformation of the original image.


Cipher Image
Pixel Substitution using Blum Blum Shub Algorithm
Intermediate Cipher Image
Column Distortion using Linear Feedback Shift Register
Row distorted Image
Row Distortion Using Linear Feedback Shift Register Algorithm
Cipher Image

4. RESULTS

4.1 Test Case I

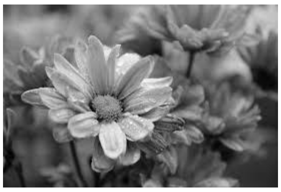


Fig1.Original Image

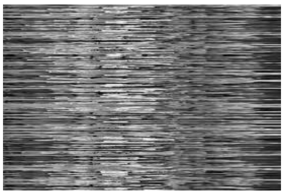


Fig2.Row Distorted Image

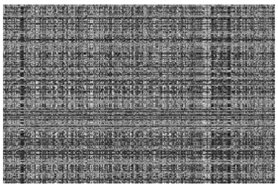


Fig3.Intermediate Cipher Image

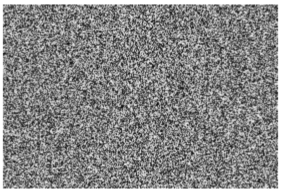


Fig4. Cipher Image

4.2 Test Case II

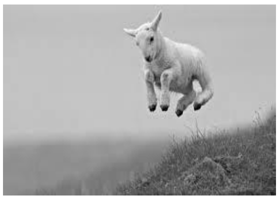


Fig1.Original Image

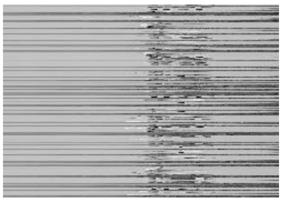


Fig2.Row Distorted Image

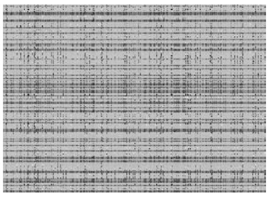


Fig3.Intermediate Cipher Image

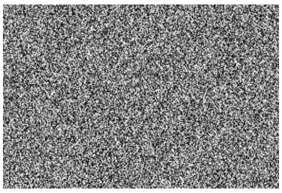


Fig4.Cipher Image

5. ANALYSIS

5.1 Correlation Coefficient

This test is used to compare the relationship between two adjacent pixels in the cipher image. Correlation coefficients among pixels of the encrypted image should be as low as possible. By arbitrarily selecting P pixels of the image, the correlation coefficient is computed by

$$r = \frac{cov(x,y)}{\sqrt{D(x)D(y)}},$$

Where

$$cov(x,y) = \frac{1}{s}\sum_{i=1}^{s}(x_i - \bar{x})(y_i - \bar{y}),$$

$$D(x) = \frac{1}{s}\sum_{i=1}^{s}(x_i - \bar{x})^2$$

5.2 Entropy

The entropy H (m) of a message source m can be measured by:

$$H(m) = \sum_{i=0}^{m-1} x^k p(m_i)\left\{log\middle|\frac{1}{p(m_i)}\right\}$$

Where M is the total number of symbols mi ∈m and p(mi) represents the probability of occurrence of symbol mi.

For all images used in our technique, after the encryption procedure, the entropy has assumed values varying from 7.9993 to 7.99941. This means that the ciphered-images are nearby to a random source and the proposed algorithm is secure against the entropy attack.

5.3 PSNR

The PSNR block calculates the peak signal-to-noise ratio, in decibels, between two images. This ratio is often used as a measurement of quality between the original and a ciphered image. The higher the PSNR, the better is the quality of the reconstructed image. PSNR is most easily defined via the mean squared error (MSE). Given a noise free m×n monochrome image I and its noisy approximation K, MSE is defined as:

$$\mathrm{MSE} = \frac{1}{m\,n}\sum_{i=0}^{m-1}.\sum_{j=0}^{n-1}[I(i,j)-K(i,j)]^2$$

$$PSNR = 10.\log_{10}(\frac{MAX_I^2}{MSE})$$

$$= 20.\log_{10}(\frac{MAX_I}{\sqrt{MSE}})$$

$$= 20.\log_{10}(MAX_I) - 10.\log_{10}(MSE)$$

Typical values for the PSNR in cipher image are between 25 and 40.

Test Case	**Correlation Coefficient**	**Entropy**	**PSNR**
	0.00383268114885	7.99930	28.1309
	0.00372942282812	7.99941	27.1225

6. CONCLUSION

As discussed earlier, secure data transmission is the need of the hour. In this paper, the authors have tried to achieve secure transmission of images by using two levels of encryption. These levels cause complete rearrangement of the pixels thereby transforming the image into an unintelligible form.

To prove the efficiency of the above technique various analysis tests are performed. The results obtained verify that the encrypted image obtained is less prone to various cryptanalysis attacks. The algorithms used are also flexible in the sense that they can be used for text and audio encryption as well.

As a part of future work, the authors recommend more secure image encryption algorithms like DES and AED. Alternatives can be looked into that can replace the initial input seed with a true random number such as CPU time or keystroke timing patterns or mouse movements. A third level of encryption could also be incorporated to make the image encryption process more secure.

7. REFERENCES

[1] Behrouz A. Forouzan, “Data Communications and Networking”, 4th Edition

[2] William Stallings, ”Cryptography and Network Security”, 5th Edition

[3] Ayushi, “A Symmetric Key Cryptographic Algorithm”, International Journal of Computer Applications (0975-8887) Volume 1- No. 15

[4] Arihant Kr. Banthia, Namita Tiwari, “Image Encryption using Pseudo Random Number Generators”, International Journal of Computer Applications (0975-8887) Volume 67- No.20, April 2013

[5] D R Stinson, “Cryptography Theory and Practice”, 3rd Edition.

[6] Faheem Masoodi, Shadab Alam, M U Bokhari, “An Analysis of Linear Feedback Shift Registers in Stream Ciphers”, International Journal of Computer Applications (0975-8887) Volume 46-No.17, May 2012

[7] M.Sahithi, B.Murali Krishna, M.Jyoti, K.Purnima, A.Jhansi Rani,N.Naga Sidhu, “Implementation of Random Number Generator Using LFSR for High Secured Multipurpose Applications”, International Journal of Computer Science and Information Technologies, Vol. 3(1),3287-3290.

[8] Nishith Sinha, Anirban Bhowmick, Kishore B, “Encrypted Information Hiding using Audio Steganography and Audio Cryptography”, International Journal of Computer Applications(0975-8887) Volume 112-No. 5,February 2015.

[9] Shrija Somaraj, Mohammed Ali Hussain, “Securing Medical Images by Image Encryption using Key Image”, International Journal of Computer Applications(0975-8887) Volume 104-No. 3, October 2014.

[10] Reshu Choudhary, Arjun JB, “Multimedia Content Security using Image Encryption”, International Journal of Computer Applications(0975-8887)

[11] Musheer Ahmad, M. Shamsher Alam, “A New Algorithm of Encryption and Decryption of Images Using Choatic Mapping”, International Journal on Computer Science and Engineering, Vol.2(1),2009,46-50.