AIVP ADVANCES IN IMAGE AND VIDEO PROCESSING
SOCIETY FOR SCIENCE AND EDUCATION UNITED KINGDOM
VOLUME 2, ISSUE 3
ISSN: 2054 - 7412

A Novel Framework for Information Hiding and Image Encryption using Least Significant Bit Techniques

H.B.Kekre[1], Tanuja Sarode[2], Pallavi N. Halarnkar[3]
[1,3]MPSTME, NMIMS University, Mumbai;
[2]TSEC, Mumbai University, Mumbai;
Hbkekre@yahoo.com, Tanuja0123@yahoo.com, Pallavi.halarnkar@gmail.com

ABSTRACT

Security of Digital Images is utmost important. Nowadays not only data but also digital images are increasing in a huge number. Good encryption techniques are always in need. In this paper we have proposed a Novel Framework, which is a combination of Information Hiding and Image Scrambling. The proposed framework is one of its kind and doesn't resemble to any existing frameworks in literature. For Image scrambling R-Prime Shuffle technique has been used.

Keywords: Image security, Information Hiding, Steganography, Image Encryption, Image Scrambling.

1 INTRODUCTION

Data security is important, so is the security of digital images. Some of the existing data security techniques cannot be directly used for securing images. Digital images have huge amount of data which makes some of the existing techniques unsuitable for securing them.

Jarno[1]proposed a modification to the existing least significant matching algorithm. In the existing technique a choice whether to add or subtract 1 is random, however the proposed method uses the choice to set a binary function of two cover pixels to the desired value. The pair of pixels is used as a unit for embedding purpose, the LSB of the first pixel carries one bit of information, and a function of two pixel values carries another bit of information. The proposed method gives the same payload as LSB but with fewer modifications to cover image. The proposed method shows a good performance as compared to LSB in terms of distortion and steganalysis attack.

A novel technique for data hiding based on Fibonacci is proposed in [2]. The method is based on bit plane decomposition for embedding the message. The technique is compared with traditional LSB method for hiding capacity.

DOI: 10.14738/aivp.23.265
Publication Date: 9th June 2014
URL: http://dx.doi.org/10.14738/aivp.23.265

Sandipan et al. proposed a data hiding technique[3]using the concept of prime numbers, which is an improvement over Fibonacci LSB method. The technique is based on decomposition of a number in sum of prime numbers. This decomposition generates a different set of virtual planes, suitable for embedding purpose. The proposed technique allows embedding in higher bit planes without much distortion and a good quality stego image is obtained. A comparative analysis between LSB, Fibonacci LSB and the proposed technique is been done. The proposed technique has proved the quality of the stego image to be much better than the other two methods.

Kekre et al. [4] proposed a steganography technique using the concept of Parity. The number of 1's and 0's are balanced by the technique in a such a way that it minimizes the possibility of suspicion. Depending upon the message bit 0/1 , the cover image byte is either modified or kept the same so as the embedding of the message bit should result in the even parity of the cover image byte.

Clandestine Data Entrenching and Salvaging, a information hiding technique was proposed by Kekre et al. [5]. In this paper two techniques are proposed, LSB 2 bit and LSB 3 bit, In the first technique , LSB and next to LSB bit are XORed, depending on the message bit 1/0 the LSB is either changed or kept the same. In the second technique, LSB bit, Next to LSB bit and Next to Next LSB bit is XORed and depending on the value of the message bit LSB is either modified or kept the same.

Information hiding technique is not limited to spatial domain, but they are explored in transform domain also. A LSB steganography technique in DCT domain is proposed by Rajib et al. in [6]. A 8x8 block of DCT coefficients is selected in cover image for embedding the secret message. A variable bit operation is applied to the selected DCT coefficients to embed a byte of secret data, where the variable bit operation is dependent on the value of the pixel. Statistical analysis is performed which shows the method is robust against various steganalysis methods.

Security of digital images is very important , a number of scrambling techniques are proposed which make the visual appearance of the digital image meaningless to the user. Until the method of scrambling is known, the user cannot descramble the digital image for its actual content. One such technique based on extension to queue transformation is proposed by Hai-Yan in [7]. The existing queue transformation technique has many disadvantages which are been overcome in this extension been proposed. The algorithm requires only one step compared to two steps to complete the scrambling. The reference point can change in every stage. To decode the image, the step, reference point all have to be known.

Image scrambling technique based on Arnold transformation is proposed by Min Li et.al in [8]. The proposed technique improves the security of image during transmission. The traditional method based on Arnold applies only to a square area, which is a limitation. This limitation has

been overcome in the proposed method by dividing the image into multiple square areas and applying the transformation for scrambling the image.

Kekre et al. proposed an Image scrambling method using the concept of Relative Prime called as R-Prime shuffle technique for Image scrambling in [9]. The method makes use of correlation concept between the rows and columns of the image. In image scrambling it is required that the correlation between the image rows and columns be minimum so as to make the details of the image unavailable to the user.

The R-Prime shuffle technique was further extended by Kekre et al. on Image blocks [10]. This method was compared to Original R-Prime shuffle on image as a whole. The method is difficult to decode as compared to original method as every block of the image has a different prime rows and columns considered for shuffling based on their correlation.

Tan Yongjie et al. [11] proposed a new method for evaluating the degree of scrambling using the grey relation analysis theory. The scrambled image is firstly analyzed so also its histogram. The scrambled image is then sub divided into sub images to construct some histogram sequences and make them small sequences. The grey relevancy of every two sequences using grey relation analysis is calculated to evaluate the image scrambling degree.

Zhang et al. proposed a Digital image encryption technique based on chaos and improved DES in [12]. The method makes use of Logistics chaos sequencer to generate pseudo random sequences, this sequence is carried on RGB image chaotically , then a double time encryption with improved DES is applied . Analysis indicate that the method is sensitive to initial condition and has high security and encryption speed.

Mohammed et al.[13] proposed a Image encryption scheme using Lagrange-Least squares Interpolation. The method consists of two main parts, Encryption/Decryption and ciphered key. The XOR operator is used in the diffusion stage to modify the pixel value which is spread to all the pixels in the image. In the substitution stage two encryption processes are used, Lagrange Process and Least square process. The decryption is just the reverse of the encryption method. The proposed system makes use of a key of length 192 bits (24 bytes). The key is expanded using AES-192 key expansion algorithm. The second approach makes use of an image as a key to cipher the plane image and the key used is expanded using CBI key expansion algorithm.

Various Encryption schemes have been compared based on a number of parameters like tunability , visual degradation, compression friendliness, format compliance , encryption ratio, speed and cryptographic security in [14]. The encryption techniques compared are from spatial as well as frequency domain. The conclusion says that none of techniques satisfies all the considered parameters for comparison.

An image security technique using the permutation of RGB components is proposed in [15]. Data bits from a textual message are encrypted using some key to some suitable non linear pixels and bit positions about the entire image. The resultant is a watermarked image. After this, three different image shares using any two components of R G and B of the watermarked image is formed. Similarly the key is also divided into three different shares and assigned to image shares.

A hybrid approach for Image security combining image encryption and steganography is proposed by Jaspal Kaur et al. in [16]. The image is firstly encrypted using modified AES algorithm then it is hidden into the cover image using the concept of steganography. The experimental results show that the hybrid method provides greater security against attacks.

Kekre et al.[17] proposed a Digital encryption method using the Random discrete distributions and MOD operator. In this paper a new parameter called as PAFCPV(Peak Average Fractional Change in Pixel Value) is proposed which gives an analysis of how good is the encryption method. The range of this parameter is between 0 to 1, where 0 means the image value is not disturbed and 1 indicates the every bit of the pixel value is affected by the encryption method.

Somdip Dey proposed a method called as SD-AEI which is an improvement over SD-EI in [18]. In the first stage every pixel value is converted to binary, equivalent to the length of the password the bits are rotated and then reversed. In the second stage the extended hill cipher technique is applied by using involuntary matrix which is generated by the same password as the first stage. In the final stage the whole image is randomized using Modified MSA Randomization encryption technique; this randomization is dependent on a unique number. The proposed method is very effective in encrypting any type of images.

2 PROPOSED INFORMATION HIDING FRAMEWORK

In this paper, a Novel Information Hiding framework is proposed. The steps of the framework are as follows

1) Take the cover image of size x bytes
2) Apply the scrambling Algorithm,(R-Prime shuffle), a scrambled image is obtained
3) To hide the message image use LSB technique(LSB 1 Bit, 2 Bit, 3 Bit and Parity), a scrambled stego image is obtained
4) The scrambled stego image obtained in step 3, apply descrambling algorithm to get a innocent stego image.

Note: The message image is hidden in the scrambled image, this scrambled stego image is now descrambled so as to obtain an innocent stego image which can be transferred across the network. Now some intruder tries to intercept the message from the innocent stego image, he

will obtain an encrypted message image which will be difficult to decrypt and get some useful information out of it. This can be seen from the experimental results obtained below.

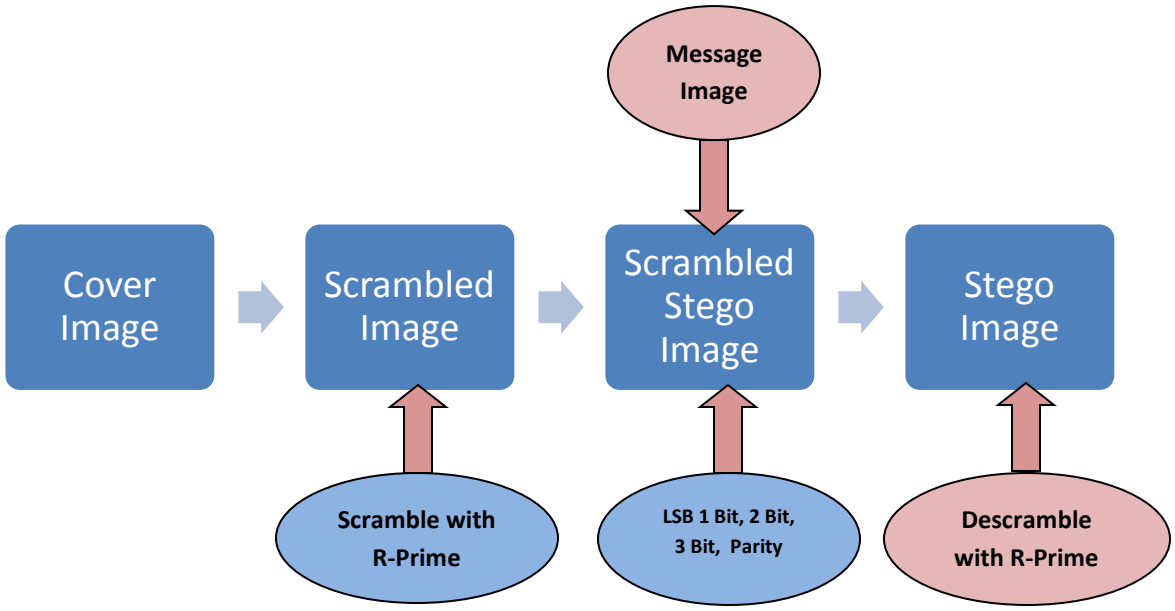


Figure 1. Embedding Stage

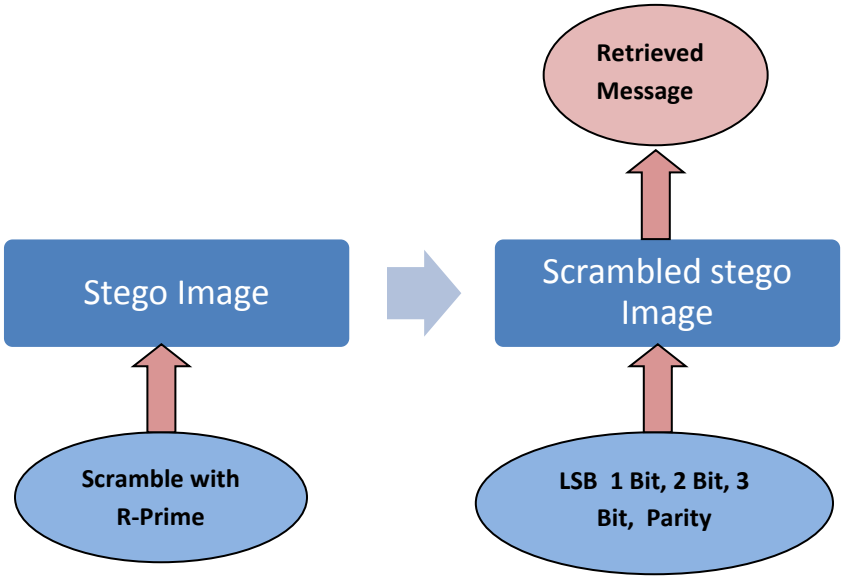


Figure 2. Retrieving Stage

2.1 R-Prime Shuffle [9]

Scrambling

The method used for Encryption is as follows

- Read the image

- Based on the Size of the Image(MXN), find out all the Relative Prime Numbers and save them in a set S
- Using set S to find the correlation of the First row with remaining rows (positions w.r.t elements present in the set).
- Consider the lowest correlation as the key to shuffle the rows in the image
- Continue till all the positions in the image are considered
- Save the Relative Primes Numbers as a key considered for Row and Column Shuffling

Repeat the same procedure for Column shuffling

Descrambling

- Use the Saved key for Row and Column Shuffling to get the Original Image back
- Use the column Relative Prime and rearrange the columns, this will give row shuffled image
- Using this row shuffled image and the key for row relative prime rearrange the rows which will give you Original Image back.
- Continue till all the positions in the image are rearranged

2.2 LSB 1-Bit Technique

Embedding Algorithm

- Take message image with size x bytes
- Cover image size should be at least 8 times x.
- Extract RGB components of pixel intensity values of message and the cover image
- Take the successive R, G, and B component values of pixels and convert them into array of values for messages and the cover image.
- Convert every decimal value into 8 bit binary equivalent for cover and message images.
- Every message bit is embedded into LSB's of the cover image.

Retrieving Algorithm

The message retrieving is done as per the algorithm given below

- Take stego image with size x bytes
- Extract R, G, B Components of pixel intensity values of stego image.
- Take successive R, G, B component values of pixels and convert them into array of values for stego image.

- Convert every decimal value into 8 bit binary equivalent for stego image.
- Retrieval of the message bit is done by extracting the LSB of every pixel from the Stego image of the R, G and B component.

2.3 Using 2 LSB's and Using 3 LSB's [5]

The technique implemented in this section just does not replace the LSB, but the LSB is modified by taking into consideration the data bits of the message, the cover image LSB and the other bits of the cover image as well. Advantages of this method are its encoding / decoding complexity is less, cover capacity is same as LSB, accuracy of retrieval is 100%, and good perceptual transparency of cover image.

Embedding Algorithm

- Take message image with size x bytes
- Cover image size should be at least 8 times x.
- Extract RGB components of pixel intensity values of message and the cover image
- Take the successive R, G, and B component values of pixels and convert them into array of values for messages and the cover image.
- Convert every decimal value into 8 bit binary equivalent for cover and message images.
- Every message bit is embedded into LSB's of the cover image after processing.
- Processing is done as follows
- If the message bit to be embedded is 0, then adjust the LSB such that the XOR ing of LSB and next to LSB is 0 and if the message bit to be embedded is 1, then adjust the LSB such that the XOR ing of LSB and next to LSB is 1 if LSB-2 bit method is being used
- If the message bit to be embedded is 0, then the LSB is adjusted such that the XOR ing of LSB next to LSB and next to next to LSB is 0. And if the message bit to be embedded is 1, then adjust the LSB such that the XOR ing of LSB next to LSB and next to next to LSB is 1 if LSB-3 bit method is being used
- Convert every 8 bits into byte for the cover image
- Take 3 bytes and group them as 3 RGB components of a 1 pixel. Perform this step for the full cover image.
- The message embedding in the cover image is over.

Retrieving Algorithm

The message retrieving is done as per the algorithm given below

- Take stego image with size x bytes
- Extract R, G, B Components of pixel intensity values of stego image.
- Take successive R, G, B component values of pixels and convert them into array of values for stego image.
- Convert every decimal value into 8 bit binary equivalent for stego image.
- Retrieval of the message bit is done by XOR ing the LSB and Next to LSB. If it is 1 then message bit is 1. If it is 0 then message bit is 0. For LSB-3 bit method, Next to Next to LSB is XOR ed.
- Convert every 8 bits into byte for the message.
- Take 3 bytes and group them as 3 RGB components of 1 pixel. Perform this step for the full message
- The message retrieval is over

Table No 1. Truth Table for LSB 2 Bit Method

LSB	Next to LSB	Message Bit	LSB Adjusted
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Table No 2. Truth Table for LSB 3 Bit Method

Next to Next LSB	Next to LSB	LSB	Message Bit	LSB Adjusted
0	0	0	0	No change
0	0	1	0	0
0	1	0	0	1
0	1	1	0	No change
1	0	0	0	1
1	0	1	0	No Change
1	1	0	0	No Change
1	1	1	0	0
0	0	0	1	1
0	0	1	1	No Change
0	1	0	1	No Change
0	1	1	1	0
1	0	0	1	No Change
1	0	1	1	0
1	1	0	1	1
1	1	1	1	No Change

2.4 Considering Parity [4]

For embedding any message image with size x bits, the cover image size should be at least x bytes. Extract the R, G, B components of the pixels, and each byte will be used to embed 1 secret message bit. Every bit of the message to be embedded is taken one at a time. To embed this bit one byte of the cover image is taken. Depending upon the value of the message bit to be embedded is 0, the LSB of the cover image byte is modified or kept same such that the parity of the cover image byte after this message bit is embedded is even. Also if the message bit to be embedded is 1, then the LSB of the cover image byte is modified or kept same such that the parity of the cover image byte after this message bit is embedded is odd

For retrieving the message the stego image(image which contains the embedded message) is taken. The parity of every byte is checked. If the parity is even that means the message bit is 0 and if the parity is odd it means the message bit is 1.

In this way after 8 such message bits are retrieved they are converted to decimal and this decimal number becomes the intensity value of the first message pixel. This procedure is repeated until the full message is retrieved, and the message image is formed.

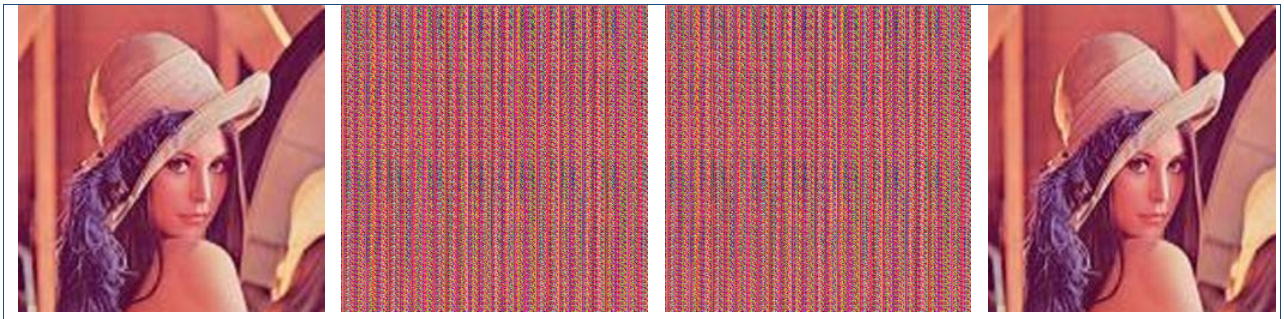
3 EXPERIMENTAL RESULTS

For Experimental purpose , five different 24-bit color images of size 256X256 were used for all the four methods , LSB 1-Bit, LSB 2-Bit, LSB 3-Bit and LSB Parity.

3.1 LSB 1-Bit

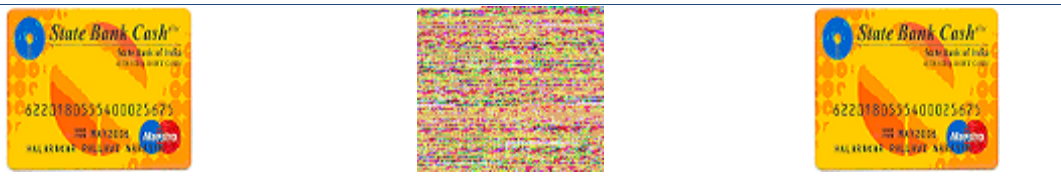
The results obtained from the proposed framework for LSB 1 Bit and displayed below. Figure 3(a) shows the original Image, 3(b) shows the scrambled image obtained after applying R-Prime Shuffle Technique, 3(C) shows the scrambled stego image obtained after embedding atm image using LSB 1 Bit method. This scrambled stego image is now converted to innocent stego image by descrambling which is seen in 3(d).

The advantage of the proposed framework can be seen from Figure.4. Figure 4(a) shows the original message image, 4(b) shows the encrypted message image obtained if some intruder tries to get it from the innocent stego image. 4(c) show the retrieved message image from the scrambled stego image.



(a) Original Image | (b)Scrambled Image | (c)Scrambled Stego Image | (d)Stego Image RMSE: 0.686

Figure. 3



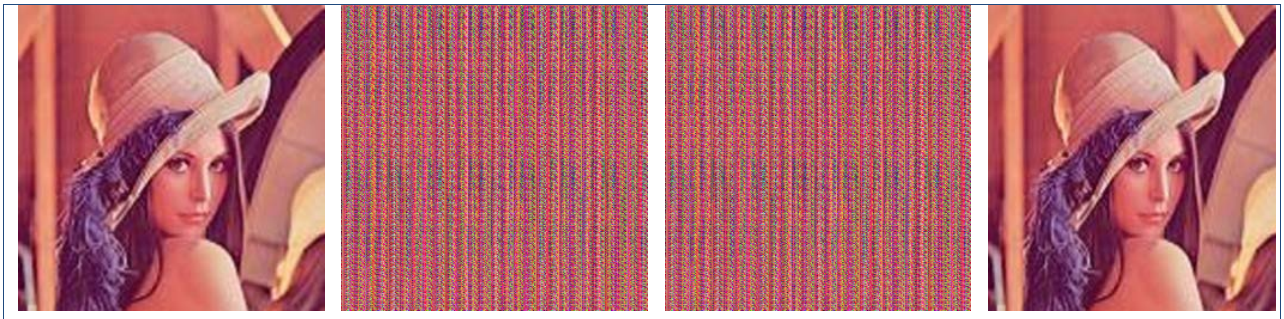
(a)Message Image | (b)Encrypted Message Image | (c)Retrieved Message Image RMSE: 0

Figure. 4

3.2 LSB 2-BIT

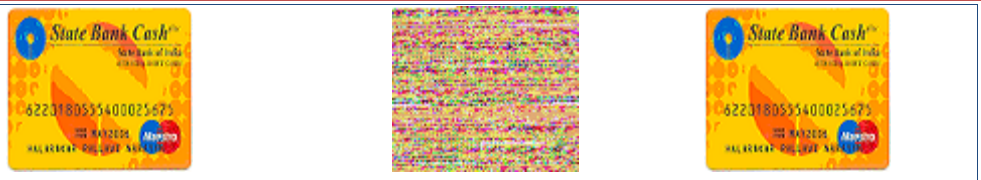
The results obtained for LSB 2 Bit are displayed below. Figure 5(a) shows the original Image, 5(b) shows the scrambled image obtained after applying R-Prime Shuffle Technique, 5(c) shows the scrambled stego image obtained after embedding atm image using LSB 2 Bit method. This scrambled stego image is now converted to innocent stego image by descrambling which is seen in 5(d).

Figure 6(a) shows the original message image, 6(b) shows the encrypted message image obtained if some intruder tries to get it from the innocent stego image. 6(c) show the retrieved message image from the scrambled stego image.



(a)Original Image | (b)Scrambled Image | (c)Scrambled Stego Image | Stego Image RMSE: 0.687

Figure 5.



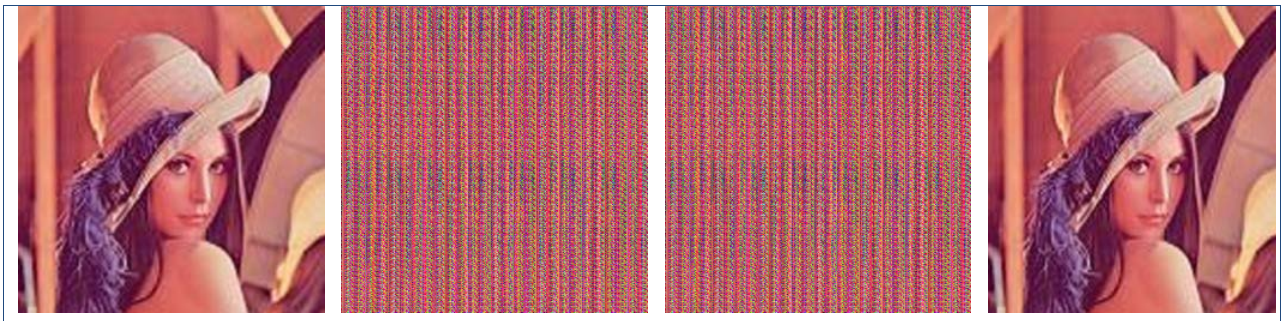
(a)Message Image (b)Encrypted Message Image (c)Retrieved Message Image RMSE: 0

Figure 6

3.3 LSB -3 BIT

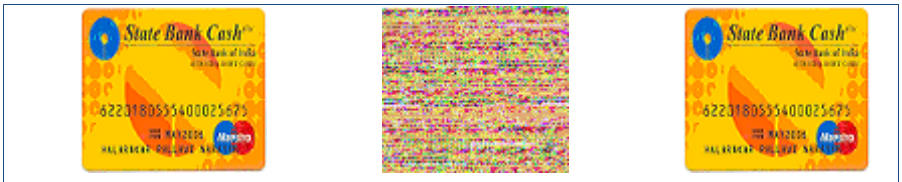
The results obtained for LSB 3 Bit and displayed below. Figure 7(a) shows the original Image, 7(b) shows the scrambled image obtained after applying R-Prime Shuffle Technique, 7(c) shows the scrambled stego image obtained after embedding atm image using LSB 3 Bit method. This scrambled stego image is now converted to innocent stego image by descrambling which is seen in 7(d).

Figure 8(a) shows the original message image, 8(b) shows the encrypted message image obtained if some intruder tries to get it from the innocent stego image. 8(c) show the retrieved message image from the scrambled stego image.



(a)Original Image (b)Scrambled Image (c)Scrambled Stego Image (d)Stego Image RMSE: 0.685

Figure 7.



(a)Message Image (b) Encrypted Message Image (c)Retrieved Message Image RMSE:0

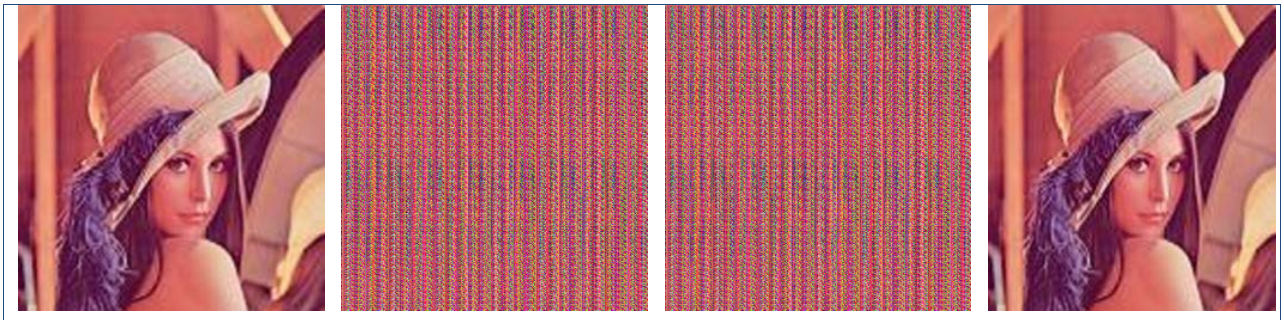
Figure. 8

3.4 LSB PARITY

The results obtained for LSB Parity and displayed below. Figure 9(a) shows the original Image, 9(b) shows the scrambled image obtained after applying R-Prime Shuffle Technique, 9(c) shows the scrambled stego image obtained after embedding atm image using LSB Parity

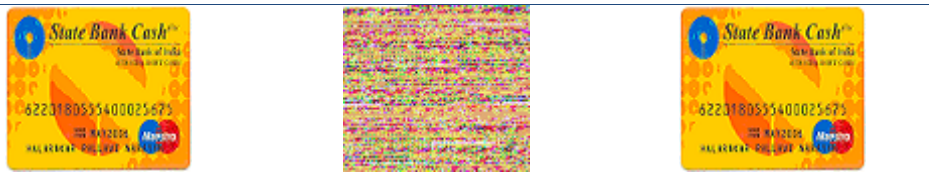
method. This scrambled stego image is now converted to innocent stego image by descrambling which is seen in 9(d).

Figure 10(a) shows the original message image, 10(b) shows the encrypted message image obtained if some intruder tries to get it from the innocent stego image. 10(c) shows the retrieved message image from the scrambled stego image.



(a)Original Image (b)Scrambled Image (c)Scrambled Stego Image (d) Stego Image RMSE: 0.686

Figue. 9



(a)Message Image (b)Encrypted Message Image (c)Retrieved Message Image RMSE:0

Figure. 10

Table No 3 gives the values obtained for Average Correlation between rows and columns of five scrambled images of size 256X256 for LSB 1-Bit, LSB 2-Bit, LSB 3-Bit and LSB Parity method. The original image correlation is also given for the respective images. It can be observed that for all the images and different LSB methods , the correlation is reduced. The table also displays values obtained for Average Moving Distance Maximum(AMD- Max) for an image of size 256X256,the AMD obtained by the scrambling technique R-Prime shuffle and Distance scrambling factor[8].

Table 3.Values of Average correlation between rows and columns of original image and scrambled image, Average Moving Distance and Distance Scrambling factor between original image and scrambled image

Lena Image Rows: 0.8454 Cols: 0.7005	LSB 1 Bit	LSB 2 Bit	LSB 3 Bit	LSB Parity
Avg Corr Rows	0.306			
Avg Corr Cols	0.268			
AMD(Max)	360.62			
AMD	136.48			
DSF	0.378			

Kutub Image Rows: 0.2784 Cols: 0.3896	LSB 1 Bit	LSB 2 Bit	LSB 3 Bit	LSB Parity
Avg Corr Rows	0.181			
Avg Corr Cols	0.187			
AMD(Max)	360.62			
AMD	133.58			
DSF	0.370			

Vegetable Image Rows: 0.5176 Cols: 0.4994	LSB 1 Bit	LSB 2 Bit	LSB 3 Bit	LSB Parity
Avg Corr Rows	0.199			
Avg Corr Cols	0.190			
AMD(Max)	360.62			
AMD	132.10			
DSF	0.366			

Baboon Image Rows: 0.4355 Cols: 0.5276	LSB 1 Bit	LSB 2 Bit	LSB 3 Bit	LSB Parity
Avg Corr Rows	0.185			
Avg Corr Cols	0.192			
AMD(Max)	360.62			
AMD	130.44			
DSF	0.361			

Fruits Image Rows: 0.6203 Cols: 0.6317	LSB 1 Bit	LSB 2 Bit	LSB 3 Bit	LSB Parity
Avg Corr Rows	0.196			
Avg Corr Cols	0.213			
AMD(Max)	360.62			
AMD	133.26			
DSF	0.369			

Table No 4 gives the values of RMSE, PSNR obtained for stego image using LSB 1-Bit , LSB 2-Bit, LSB 3-Bit and LSB Parity.

Table 4. Values of RMSE, PSNR, between original Image and Stego image

Lena	**LSB 1 Bit**	**LSB 2 Bit**	**LSB 3 Bit**	**LSB Parity**
RMSE	0.686	0.687	0.685	0.686
PSNR	51.39	51.38	51.40	51.40
Kutub	**LSB 1 Bit**	**LSB 2 Bit**	**LSB 3 Bit**	**LSB Parity**
RMSE	0.685	0.685	0.685	0.684
PSNR	51.41	51.40	51.41	51.42
Vegetable	**LSB 1 Bit**	**LSB 2 Bit**	**LSB 3 Bit**	**LSB Parity**
RMSE	0.684	0.687	0.686	0.685
PSNR	51.42	51.39	51.40	51.41
Baboon	**LSB 1 Bit**	**LSB 2 Bit**	**LSB 3 Bit**	**LSB Parity**
RMSE	0.685	0.687	0.686	0.686
PSNR	51.40	51.38	51.40	51.39
Fruits	**LSB 1 Bit**	**LSB 2 Bit**	**LSB 3 Bit**	**LSB Parity**
RMSE	0.684	0.686	0.686	0.687
PSNR	51.41	51.40	51.40	51.38

Table No 5 gives the values of PAFCPV and NPCR obtained for encrypted message image using LSB 1-Bit , LSB 2-Bit, LSB 3-Bit and LSB Parity

Table 5. Values of PAFCPV and NPCR, between original Message Image and Encrypted Message image

Lena	**LSB 1 Bit**	**LSB 2 Bit**	**LSB 3 Bit**	**LSB Parity**
PAFCPV	0.316	0.316	0.316	0.316
NPCR	99.34	99.35	99.34	99.36
Kutub	**LSB 1 Bit**	**LSB 2 Bit**	**LSB 3 Bit**	**LSB Parity**
PAFCPV	0.306	0.305	0.306	0.305
NPCR	97.91	97.83	97.87	97.86
Vegetable	**LSB 1 Bit**	**LSB 2 Bit**	**LSB 3 Bit**	**LSB Parity**
PAFCPV	0.397	0.398	0.398	0.397
NPCR	99.39	99.39	99.38	99.40
Baboon	**LSB 1 Bit**	**LSB 2 Bit**	**LSB 3 Bit**	**LSB Parity**
PAFCPV	0.297	0.297	0.297	0.296
NPCR	98.89	98.88	98.86	98.88
Fruits	**LSB 1 Bit**	**LSB 2 Bit**	**LSB 3 Bit**	**LSB Parity**
PAFCPV	0.325	0.325	0.325	0.324
NPCR	99.09	99.11	99.08	99.08

4 OBSERVATIONS

It can be observed from the experimental results the maximum reduction in the correlation for rows is obtained in fruits image where the correlation is reduced by 68.41 %. Minimum column correlation is obtained in fruits image with a reduction in column correlation by 66.29%. DSF is maximum for Lena image. Mean squared error obtained for all the stego images is approximately the same so also is the PSNR for all the methods of LSB. The encrypted message image obtained for Vegetable image gives a good value for PAFCPV. The NPCR value obtained incase of Kutub Minar image is less as compared to other images.

5 CONCLUSION

In this paper, we have proposed a Novel Approach for securing the message image. A lot of Information hiding techniques are proposed in literature, they make use of existing encryption methods to encrypt the message image and then embed it in the cover image to create a stego image which can be transferred, here our framework uses completely a different approach, using a simple scrambling technique, we a getting a good encrypted message image from a stego image which will be difficult to decrypt.

REFERENCES

[1]. Mielikainen, Jarno. *LSB matching revisited*. Signal Processing Letters, IEEE 13, no. 5 (2006) .p. 285-287.

[2]. Battisti, F., M. Carli, A. Neri, and K. Egiaziarian. *A Generalized Fibonacci LSB Data Hiding Technique.* Computers and Devices for Communication (CODEC-06), Institute of Radio Physics and Electronics, University of Calcutta International Conference on. 2006.

[3]. Dey, Sandipan, Ajith Abraham, and Sugata Sanyal. *An LSB data hiding technique using prime numbers.* In *Information Assurance and Security, 2007. IAS 2007. Third International Symposium on*,. IEEE, 2007. p. 101-108

[4]. Kekre, H. B., and Archana A. Athawale. *Personal Data Ingrain and Regain., NCA, FCRCE, Bandra (W), Mumbai, 16th-17th May* (2008).National Conference on Algorithms p. 88-93

[5]. Kekre, H. B., Archana A. Athawale, and Sudeep D. Thepade. *Clandestine Data Entrenching and Salvaging.* In *National Conference on Information and Communication Technology 29th Feb and 1st Mar, NCICT-08*.p. 1-7

[6]. Biswas, Rajib, Sayantan Mukherjee, and Samir Kumar Bandyopadhyay. *DCT Domain Encryption in LSB Steganography.* In Computational Intelligence and Communication Networks (CICN), 2013 5th International Conference on,. IEEE, 2013. p. 405-408

[7]. Zhang, Hai-Yan. *A new image scrambling algorithm based on queue transformation.* In Machine Learning and Cybernetics, 2007 International Conference on, vol. 3, IEEE, 2007. p. 1526-1530.

[8]. Li, Min, Ting Liang, and Yu-jie He. *Arnold Transform Based Image Scrambling Method.* Multimedia Tecnhnology (ICMT 2013) International Conference on .2013. p.1309-1316

[9]. Kekre, H. B., Tanuja Sarode, and Pallavi Halarnkar. *Image Scrambling using R-Prime Shuffle*. IJAREEIE, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering.2013. 2(8). p.4070-4076

[10]. Kekre, H. B., Tanuja Sarode, and Pallavi Halarnkar. *Image Scrambling using R-Prime Shuffle on Image and Image Blocks.* IJARCCE. International Journal of Advanced Research in Computer and Communication Engineering. 2014. 3(2). p.5471-5476.

[11]. Yongjie, Tan, and Zhou Wengang. *Image scrambling degree evaluation algorithm based on grey relation analysis.* In Computational and Information Sciences (ICCIS), 2010 International Conference on, IEEE, 2010 p. 511-514.

[12]. Yun-Peng, Zhang, Liu Wei, Cao Shui-ping, Zhai Zheng-jun, Nie Xuan, and Dai Wei-di. *Digital image encryption algorithm based on chaos and improved DES*. In Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on, IEEE, 2009.p. 474-479.

[13]. Shreef, Mohammed A., and Haider K. Hoomod. *Image Encryption Using Lagrange-Least Squares Interpolation.* (IJACSIT). International Journal of Advanced Computer Science and Information Technology. 2013 2(4).p. 35-55.

[14]. Shah, Jolly, and Vikas Saxena. *Performance Study on Image Encryption Schemes.* (IJCSI) International Journal of Computer Science. 2011. 8(4).p.349-355.

[15]. Samanta, Sabyasachi, Saurabh Dutta, and Goutam Sanyal. *An enhancement of security of image using permutation of RGB-components.* In Electronics Computer Technology (ICECT), 2011 3rd International Conference on, vol. 2, IEEE, 2011. p. 404-408.

[16]. Saini, Jaspal Kaur, and Harsh K. Verma. *A hybrid approach for image security by combining encryption and steganography.* In Image Information Processing (ICIIP), 2013 IEEE Second International Conference on, IEEE, 2013. p. 607-611.

[17]. Kekre, H. B., Tanuja Sarode, and Pallavi Halarnkar. *Performance Evaluation of Digital Image Encryption Using Discrete Random Distributions and MOD Operator.* IOSR-JCE. IOSR Journal of Computer Engineering. 2014. 16(2).Ver V. p.54-68.

[18]. Dey, Somdip. "SD-AEI: An advanced encryption technique for images." In*Digital Information Processing and Communications (ICDIPC), 2012 Second International Conference on*, pp. 68-73. IEEE, 2012.

