Reversible Data Hiding in Encrypted Images with Private Key Cryptography

Wajahath Hussain Razvi, Dr.Ch.Samson

***Abstract*— This project proposes a reversible scheme for cipher images which are encrypted using a simple cipher involving key bunch matrix. Key bunch matrix is a simple and convenient method to encrypt an image. The secret text data should be converted into its equivalent decimal values using EBCDIC code. The decimal values are converted into binary; then the data is embedded into cipher image by using lossless steganography technique based on the key and sent to the receiver. The secret data is extracted from the received image using key. The receiver generates decryption key and decrypts the image using the same key. The scheme is able to embed secret data in encrypted image in a secure manner without any loss.**

***Index Terms*— Reversible data hiding, Image encryption, Key bunch matrix, Exclusive OR**

I. INTRODUCTION

Encryption and data hiding are two of the most effective means of data protection, while the encryption technique convert plaintext image into unintelligible/unreadable cipher text image, the data hiding technique embed additional data into cover image by slight modifications. In some places like medical, defense etc., data hiding may be performed with reversible technique so that the cover image can be recovered without any distortion.

A data hiding technique is called reversible data hiding technique if the original cover Image can be perfectly recovered from the other cover image version containing embedded data even though some modifications are done for embedding the secret data. The reversible data hiding in encrypted image is examined in Most of the work on reversible data hiding focuses on the data embedding/extracting on the plain text domain. But, in some applications, an administrator appends some secret data, such as the authentication data or request's etc., within the encrypted image without knowing the contents of the image. The original image can be recovered without any distortion after image is decrypted and secret data can be extracted at receiver side. There are many types of embedding techniques available but most of them are lossy techniques and sometimes it difficult to recover the original image.

Mr. Wajahath Hussain Razvi, B.Tech degree in the field of computer science from JNTU Hyderabad and is presently pursuing M.Tech(C.S.E) at MVSR Engineering College.

Dr. Ch. Samson, Diploma from Govt. Polytechnic, Hyderabad in 1994, B. E. from Osmania University , M. E from SRTM University Ph.D. from JNTU Hyderabad, working as Associate Professor and Head in the Dept. of Information Technology at MVSR engineering College.

II. PROPOSED TECHNIQUE

In the proposed system an input image is taken and is encrypted using a cipher involving key bunch matrix a simple and convenient encryption technique which has 128 bit keys, With respect to data hiding key the key maps to a particular pixel. The pixel is then converted into its equivalent to its 8bit binary code. The secret data is also converted into EBCDIC code which is later converted into 8bit binary number For each pixel that is selected the MSB of pixel in image is XORed with the 2bits of secret data. Again it is XORed with the LSB and stored at the LSB position of the pixel. Mostly the pixel value doesn't change; if at all it changes the real value can be recovered by this method While retrieving the data first the secret data is recovered, then the image is decrypted.

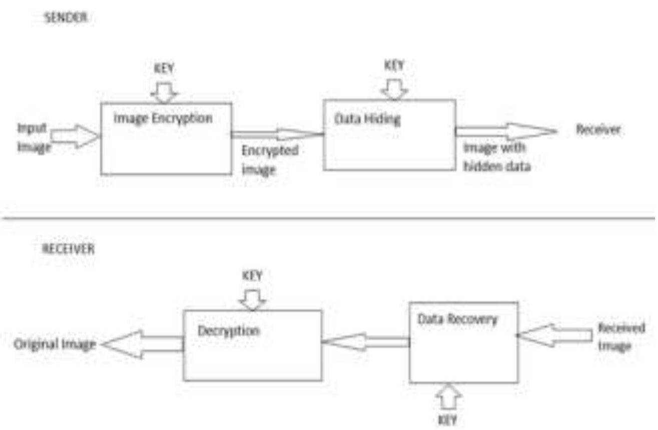

Figure 1 Architecture

III. KEY BUNCH MATRIX

A key bunch matrix is asymmetric key supported by a key-based permutation and a key-based substitution. The decryption key can be obtained by using the given encryption key bunch matrix and it is the concept of multiplicative inverse. The strength of the cipher is so good that it cannot be broken by any conventional attack as shown in [2].

Consider a plaintext 'P' which can be represented in the form of a matrix given by

$P = [\ p_{ij}\ ]$, i=1 to n, j=1 to n, (3.1) where in each 'p_{ij}' is a decimal number lying in [0-255]. Let

$E = [\ ij\ e\ ]$, i=1 to n, j=1 to n, (3.2)

be the encryption key bunch matrix, where each 'e_{ij}' is an odd number lying in [1-255], and $D = [\ d_{ij}\ ]$, i=1 to n, j=1 to n, (3.3)

is the decryption key of thebunch matrix, wherein each 'd_{ij}' is an odd number lying in [1 to 255]. ij e and d_{ij} are connected by the relation $(\ ij\ e \times d_{ij}\ ) \mod 256 = 1$, (2.4)

Here it may be noted that the dij is obtained corresponding to every given ij e in an appropriate manner. The basic equations governing the encryption and the decryption processes of the cipher can be written in the form

C = [ij c]=[ij e × pij] mod 256, i=1 to n, j = 1 to n (3.5)

And

P = [pij]=[dij × ij c] mod 256, i=1 to n, j = 1 to n. (3.6)

On assuming that the cipher involves an iteration process, the flowcharts governing the encryption and the decryption can be drawn as shown in Figs. 1 and 2.

Algorithm for Encryption

1. Read P,E,K,n,r
2. For k = 1 to r do
3. For i=1 to n do
4. For j=1 to n do
5. pij = (ij e × pij) mod 256
6. P=[pij]
7. C=P
8. Write(C)

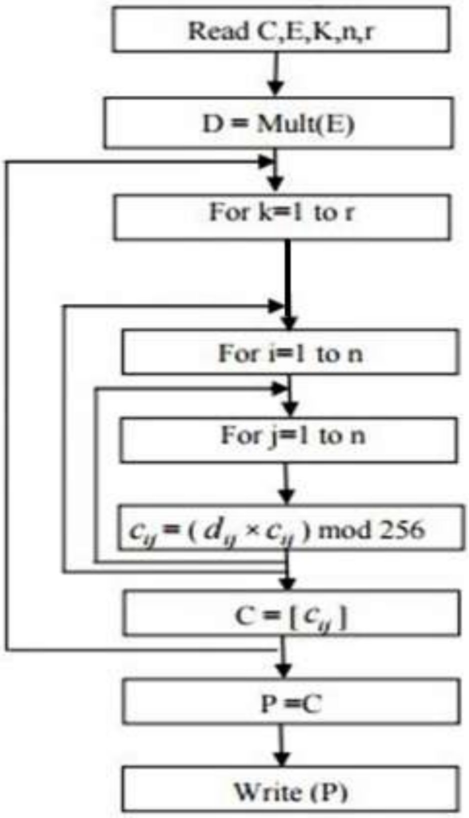

Figure 2 Encryption

Algorithm for Decryption

1. Read C,E,K,n,r
2. D=Mult(E)
3. For k = 1 to r do
4. For i =1 to n do
5. For j=1 to n do
6. ij c = (dij × ij c) mod 256
7. C=[ij c] }
8. P=C
9. Write (P).

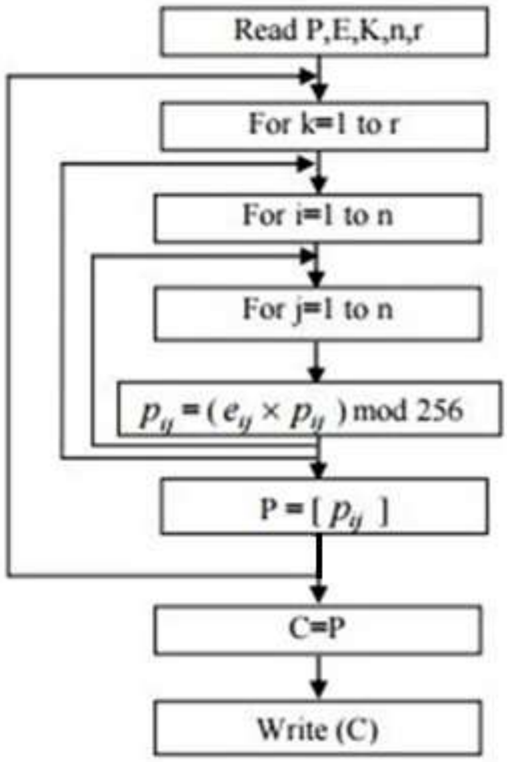

Figure 3 Decryption

IV. DATA HIDING

With respect to the key in this technique the data is inserted at Least Significant Bits. Suppose 'I' is the cover image and its pixel are generated with respect to key and after this the pixel are converted into its 8 bit binary number. This 8 bit binary number is used for further computation. By using the 8th bit of present pixel and the 1st bit of secret message, XOR operation is performed and in the same way the 7th bit and 2^{nd} bit of secret message is XORed, then it is again xor with 1^{st} and 2^{nd} bit of present pixels. These two computed bits are inserted at the last two LSBs of the present pixel. This cycle repeats until the value of the message becomes zero i.e. message ended.

Embedding Algorithm

1. Input – Cover Image (IC),Secret Message(D) and Secret Key (KKey)
2. Pick a pixel IC(x, y)that key maps from the image according to the key and convert it into 8 bit binary number.
3. Find the 8th bit and 7th bit of the pixel value.
4. Find secret message first and second bit
5. Perform XOR operation on 8^{th} bit of pixel with first bit of secret message and also perform XOR operation on 7^{th} bit of pixel with second bit of secret message.
6. Perform Xor on new bit 1 and 7^{th} bit and on 2^{nd} and 8th
7. Store the obtained bit at second and first LSB position of the present pixel.
8. Repeat step 2 for each pixel until the message is terminated.

International Journal of Engineering and Technical Research (IJETR)
ISSN: 2321-0869 (O) 2454-4698 (P) Volume-7, Issue-11, November 2017

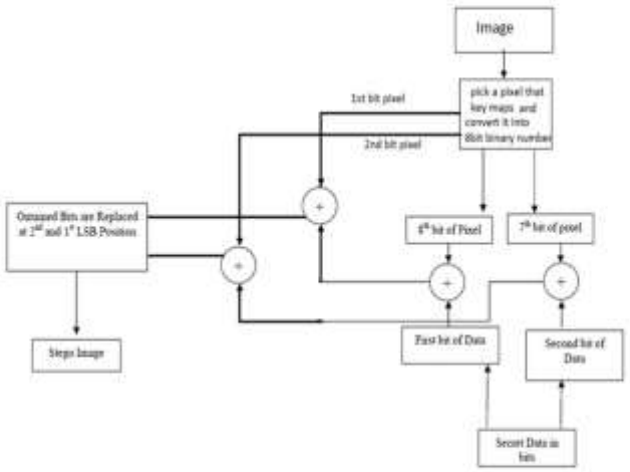

Figure 4 embedding Data

Extraction Algorithm

1. Initialize Is ←Stego image,KKey ←Secret Key
2. For each column the key maps.
3. Pick a pixel Is(x, y) from the image wrt. kkey and convert it into 8 bit binary number.
4. Find the 8th, 7th, 2nd, 1st bit of the pixel value.
5. Perform XOR on (eight,second) bit and also on (seventh, first) bit,
6. Perform XOR on the obtained(step5) 1^{st} bit , 8^{th} bit and also on 2^{nd} obtained bit , 7^{th} bit. This is the secret data bits.
7. Perform XOR on obtained(step5) secret bit (first,seventh) and (second, eight) .
8. This will give image data bits.
9. Repeat step 2 for each pixel that key maps.

Example:

Embedding:

a. Pixel 255= 11111111 and secret data 10110110
b. 1^{st} and 2^{nd} bit =11(b)
c. 7^{th} and 8^{th} bit= 11(d)
d. first two bits of secret data 10(a),
e. bitXOR(a,b)=01(c)
f. bitXOR(c,d)= 10 (e)
g. e is written in original pixel it becomes 11111110 i.e 254

Extraction:

a. 254 is converted into binary number 11111110
b. 1^{st} 2^{nd} 11(b) 7^{th} and 8^{th} 10(e)bits are taken.
c. bitXOR(b,e)= 01(c)
d. bitXOR (c,b)= 10(a)- secret data pixel
e. bitXOR (c,e)= 11(d) –original image pixel
f. So we got 11111111 i.e 255.

V. EXPERIMENTAL RESULT

A gray scale image of size 256X256 is taken to perform encyption using Key bunch matrix.

Encyption key is taken as

e=[161 157 159 189
221 209 179 111
249 227 185 177
147 239 89 243];

As we have 4X4 key so the key is repeated.

Now the receiver knowing the encryption can obtain a decryption key by calculating

$$Eij * Dij \bmod 256 = 1 \qquad (5.1)$$

Figure 5 Input image

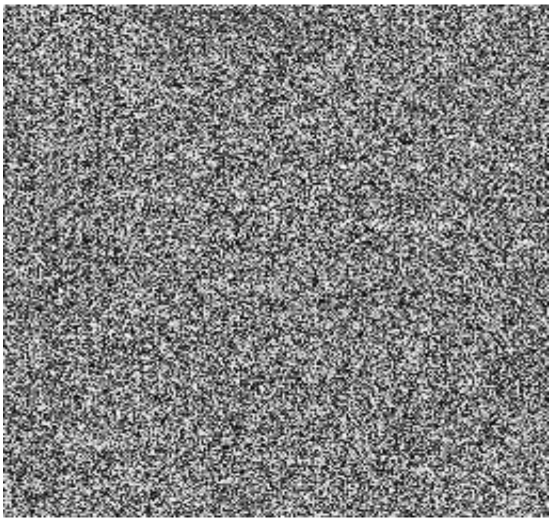

Figure 6 Encrypted Image

Now Secret data is taken

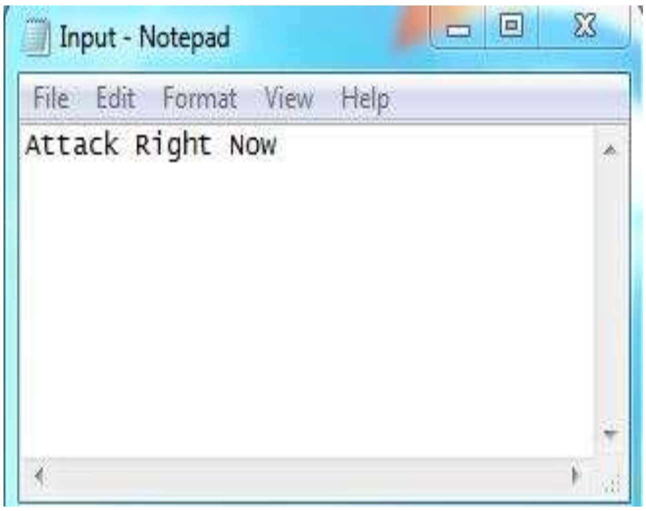

Figure 7 Input text or secet data

Secret data is embedded now

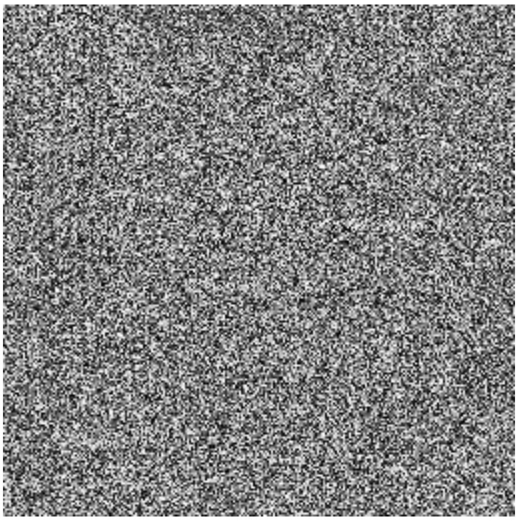
Figure 8 Secret data Embedded

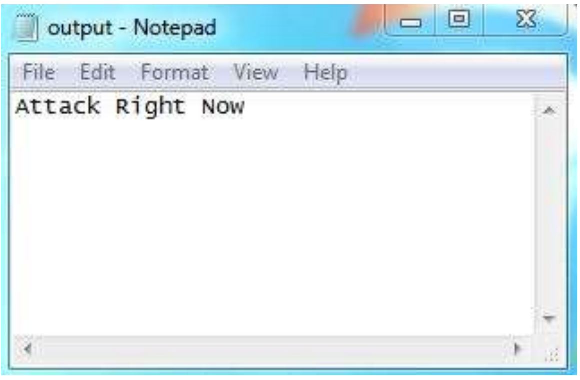

Figure 9 Extract the Secret data at receiver's side

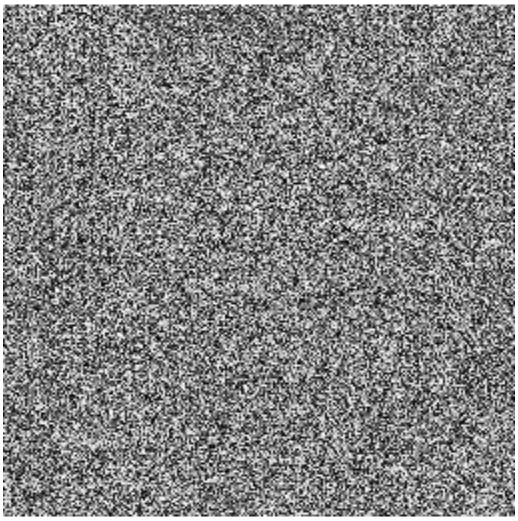
Figure 10 Recovered Encrypted image

Now, that the data is extracted from the encrypted image the encrypted image is reconstructed.

'd' is the key that can be obtained by calculating 5.1

d=[97 181 95 149
117 49 123 143
73 203 137 81
155 15 233 59];

The receiver can decrypt the image using the above key.


Figure 11 Decrypted image

VI. COMPARISION

S.no	Algorithm	Packet Size (KB)	Encryption Time (sec)	Decryption time (Sec)
1	AES	153	1.6	1
	DES	153	3.0	1.1
	RSA	153	7.3	4.9
	Proposed method	153	0.8	0.8
2	AES	196	1.7	1.4
	DES	196	2.0	1.24
	RSA	196	8.5	5.9
	Proposed method	196	1	1
3	AES	312	1.8	1.6
	DES	312	3.0	1.3
	RSA	312	7.8	5.1
	Proposed method	312	1.3	1.3
4	AES	868	2.0	1.8
	DES	868	4.0	1.2
	RSA	868	8.2	5.1
	Proposed method	868	1.7	1.7

Table 1. AES, DES,RSA and Key bunch matrix Time Comparison.

By the above table, the time taken to encrypt and decrypt the data using RSA is higher than any other algorithm.

Cipher	Key Size	Time to break	Type	Key
RSA	1024 to 4096	7481 years (1024 bit)	Public	Asymmetric Key
DES	56 bits	56 hours[5]	Private	Symmetric Key
Proposed method	128 bits	$3.12X10^{23.4}$ Years	Private	Inverse Key

Table 2 Cipher comparison

As the length of the key K is 128 binary bits. As the equations governing the cipher, are non-linear and therefore envisage that it is not possible to choose either a plaintext or a ciphertext for breaking the cipher. This cipher is a strong one and it cannot be broken by any conventional attack.

Method	Size	PSNR	MSE	Paper
Proposed method	256X256	51.7820	0.4314	
Wet paper coding	256X256	40.3	0.9231	Lossless and Reversible data hiding in Encrypted images with public key cryptography[]
XOR	256X256	47.2773	1.2172	An Enhanced Method for Data Hiding Using 2-bit XOR in Image Steganography[4] .

Table 3 Image Comparison

VII. CONCLUSION

Reversible data hiding in encrypted images using private key cryptography is implemented in this project, the cipher pixel values are replaced with new values which contains embedded data in LSB. Here, the hidden data can be extracted directly from the encrypted image by using key. The embedding operation does not affect the original image. The prevention of data attack can be reduced and Information security can be provided at greater extend. Total lossless data recovery is possible at the time of data extraction. The data embedding is lossless and also secure as it is a key based method. Image encryption is simple and provides better security.

REFERENCES

[1] N.A Saleh, H. N. Boghdad, S. I, A. M. Darwish, "High Capicity Lossless Data Embedding Technique for Palette Images Based on Histogram Analysis," Digital Signal Processing, 20, pp. 1629-1636, 2010

[2] Dr. V.U.K.Sastry, Ch Samson "Encryption of a gray level Image and a color image by using a key bunch matrix" International Journal of Advanced Computing, vol 45 June 2013, ISSN:2051-0845

[3] Xinpeng Zhang, Jing Long, Zichi Wang, and Hang Cheng, "Lossless And Reversible Data Hiding in Encrypted Images with Public key Cryptography",10,pp.1051-8215,2015

[4] Kamldeep Joshi, Rajkumar Yadav, Gaurav Chawla "An Enhanced Method for Data Hiding using 2-Bit XOR in Image Steganography" International Journal of Engineering and Technology, ISSN 0975-4024

[5] Wikipedia.orghttps://en.wikipedia.org/wiki/Data_Encryption_Standard

Mr. Wajahath Hussain Razvi received B.Tech degree in the field of computer science from JNTU Hyderabad and is presently pursuing M.Tech(C.S.E) at MVSR Engineering College, He published one paper in national conference. His research interests are image processing and network security.

Dr. Ch. Samson obtained his Diploma from Govt. Polytechnic, Hyderabad in 1994, B. E. from Osmania University in 1998, M. E from SRTM University in 2000 and Ph.D. from JNTU Hyderabad, He has published 25 research papers in international journals and two papers in conferences. He is currently working as Associate Professor and Head in the Dept. of Information Technology at MVSR engineering College. His research interests are Image Processing, Image Cryptography and Network Security