Hiding of Random Permutated Encrypted Text using LSB Steganography with Random Pixels Generator

Noor Kareem Jumaa
Department of Computer Technology Engineering
Al-Mansour University College,
Iraq

ABSTRACT

In today's world of growing technology, network security and protection of data have been of great concern. Steganography is the art of hiding a private message with in a cover file in such a manner that third party cannot know the existence of the hidden message. Steganography has many different forms like LSB, transforms, masking, and filtering technique. Enhanced LSB technique is main concern of this paper. The implementation in this paper is concern with five features. First one is deal with the generation of truly random and secure encryption key, second feature deals with encrypt the secret message using AES algorithm, third feature deals with generation of permutation technique to distribute the encrypted message bits randomly, fourth feature dealing with generation of random pixels number in order to hide the secret encrypted message bits inside them, finally the fifth features id dealing with the hiding the random ciphertext bits within random image pixels. This paper proposed an enhanced hiding system with analysis of the proposed system.

Keywords

information hiding, AES, steganography, LSB, ciphertext.

1. INTRODUCTION

The "steganography" word comes from a Greek word “stegano” which means covered or hidden and the first uses of steganography is recorded which could be traced back to 440 B.C. to communicate in wars. Then, various possible permutations are done on the used applications to obtain new and more ways of carrying out the information in an efficient way.

Steganography is the art of hiding a secret message within a cover message in such a manner that third unauthorized parity cannot know the existence of the hidden information nor the manner. Actually, steganography is a method for secret communication that is about hiding the existence of a messages, where as the classical cryptography is about hiding the content of messages. [1, 2, 3]

Steganography can be split into four main categories according to the cover medium as shown in Fig. (1): [4]

I. Text Steganography.

II. Image Steganography.

III. Video/ Audio Steganography.

IV. Protocol Steganography.

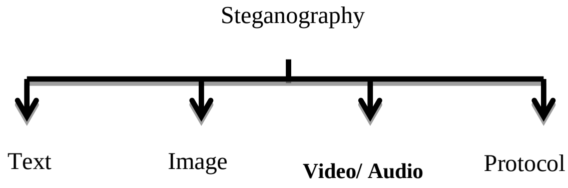


Fig. (1): Steganography Categories.

The main terms used in the steganography systems are: [5 ,6]

I. **Cover message:** is the carrier of the message such as image, video, audio, text, or some other digital media.

II. **Secret message:** is the information which is needed to be hidden in the suitable digital media.

III. **Secret key:** is usually used to embed the message depending on the hiding algorithms.

IV. **Embedding algorithm:** is the way or the idea that usually use to embed the secret information in the cover message.

Cryptography and steganography are related to each other. Steganography could be combined with cryptography, so the message cannot be read even though the message is discovered. The difference key between Cryptography and Steganography lies with the fact that when a message is encrypted, third party knows about the message existence and knows that it is encrypted, whereas with Steganography third party might not even know nor observe that there does exists a message too, simply because it is hidden inside a cover which can be an image, video or an audio etc., which without keen observation might go without any suspicion. [2, 7]

In this paper, LSB (Least Significant Bites)-image steganography is implemented.

The rest of this paper is organized as follows: section 2 provide a brief introduction about the steganography scenario system, section 3 provide a description about the LSB steganography, section 4 describe random number generator method, section 5 discuss the proposed model, results and analysis described in section 6 and section 7 contains the conclusions.

2. STEGANOGRAPHY SCENARIO SYSTEM

Before the hiding process in the steganography system scenario, the sender must choice the suitable message carrier (i.e. image, video, audio, or text) and selects the effective secret messages in addition to the robust password -which suppose to be known by the receiver-. The effective and appropriate steganography algorithm must be selected that able to encode the message in more secure technique. Then the sender may send the stego file by email or chatting, or by other modern techniques. The stego file is the carried message with the secret information. After receiving the message by the receiver, he can decode it using the extracting algorithm and the same password used by the sender. The Steganography system scenario is shown in the Fig. (2). [5]

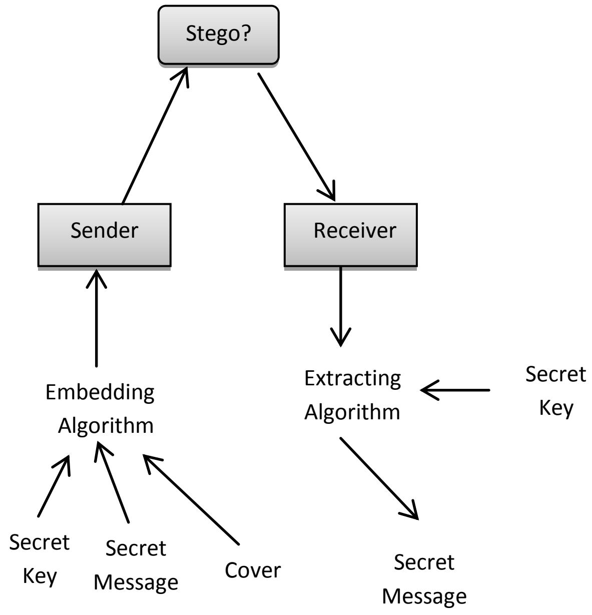


Fig. (2): Steganography System Scenario.

3. LSB IMAGE BASED STEGANOGRAPHY TECHNIQUE

Least significant bit (LSB) insertion is a common, simple approach to embedding information in a cover image.

LSB algorithm can be described as follows:

a. The cover image is read as a matrix of pixels. For example: the 2D gray scale image is converting into its equivalent pixels as shown in Fig. (3).

b. Convert the selected pixel for the hiding to its binary presentation.

   For example: if the pixel value= **193** in decimal, the binary presentation is **11000001**.

c. The secret message characters are converting to their equivalent binary forms.

   For example: if the secret message= **h**, the binary presentation is **01101000**.

d. Now, the message binary form is xoring with the pixel binary form.

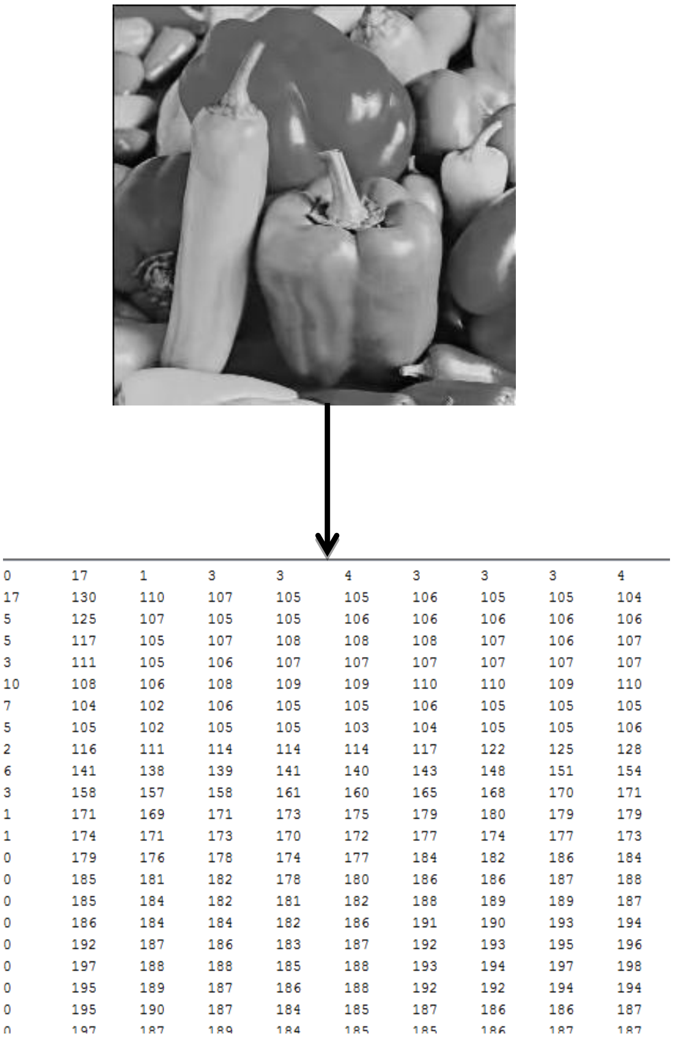
i.e. : **11000001**

XOR

01101000

10101001 ← the resulted stego pixle is **169**

Fig. (4) Shows the LSB embedding Algorithm.



0	17	1	3	3	4	3	3	3	4
17	130	110	107	105	105	106	105	105	104
5	125	107	105	105	106	106	106	106	106
5	117	105	107	108	108	108	107	106	107
3	111	105	106	107	107	107	107	107	107
10	108	106	108	109	109	110	110	109	110
7	104	102	106	105	105	106	105	105	105
5	105	102	105	105	103	104	105	105	106
2	116	111	114	114	114	117	122	125	128
6	141	138	139	141	140	143	148	151	154
3	158	157	158	161	160	165	168	170	171
1	171	169	171	173	175	179	180	179	179
1	174	171	173	170	172	177	174	177	173
0	179	176	178	174	177	184	182	186	184
0	185	181	182	178	180	186	186	187	188
0	185	184	182	181	182	188	189	189	187
0	186	184	184	182	186	191	190	193	194
0	192	187	186	183	187	192	193	195	196
0	197	188	188	185	188	193	194	197	198
0	195	189	187	186	188	192	192	194	194
0	195	190	187	184	185	187	186	186	187
0	197	187	189	184	185	185	186	187	187

Fig. (3): Image to Pixels Conversion.

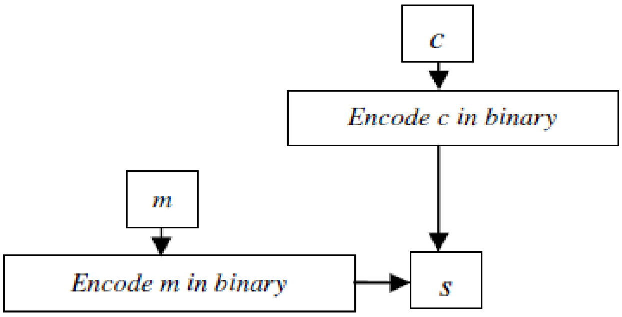


Fig. (4): LSB Embedding Algorithm. [3]

From the Fig. above, LSB hiding technique can be described as following: .[4, 8].

1. Select *cover-object c* as an input.

2. Encode the *c* in binary.

3. The Secret Message, *m*.

4. Encode the *m* in binary.

5. Choose one pixel of the *c* randomly.

6. Use a pixel selection to hide information in the *c*.

7. Save the new image (*Stego-image*) *s*.

The advantage of LSB hiding technique is its simplicity and many techniques use this method. LSB steganography art also allows high perceptual transparency. However, there are many weaknesses when robustness, tamper resistance, and other security issues are considered. There is an extremely sensitivity in LSB encoding to any type of filtering or manipulation of the stego-image. Scaling, rotation, cropping, addition of noise, or loss compression to the stego-image is very likely to destroy the message. Furthermore an attacker can easily remove the message by removing the entire LSB plane with very little change in the perceptual quality of the modified stego-image. [4, 9]

Advantages of LSB embedding technique are: [4]

1. LSB algorithms quick and easy, this is the major advantage of LSB technique.
2. Also, there has been steganography software developed which work around LSB color alterations via palette manipulation.
3. LSB insertion works well with gray-scale images.

4. PSEUDO RANDOM NUMBER GENERATOR (PRNG)

Most random number generators are in fact "pseudo-random" number generators: Not Really Random Numbers. They generate values according to some internal equations. The nature of the equations tends to be such that the values appear to be random and perhaps even pass many statistical measures of randomness. However, all "pseudo-random" number generators have a "cycle". After a single cycle has passed, some underlying property of the numbers repeats in the same order as it appeared before. [10]

Linear Feedback Shift Register (LFSR) is a PRNG, It is a feedback shift register which made up of two parts: [10]

- Shift register.
- Feedback function.

The simplest kind of feedback shift register is a linear feedback shift register (LFSR), the feedback function is simply the XOR of certain bits in the register; the list of these bits is called a **tap sequence.**

An LFSR with n flip-flops can implement only a (2n–1) state counter. The all-zeros state is normally not allowed because the counter locks up. Good design practice demands a reset condition that provides startup in a known condition and also ensures that the counter does not power up in a zero Condition and stay locked up. The choice of the polynomial used should ensure 2n–1 states—with no repeated states; such a polynomial is known as a primitive or maximal-length polynomial. [10]

The maximal length sequence has the following properties: [10]

- The number of ones in a sequence approximately equals the number of zeros.
- The statistical distribution of ones and zeros is well defined and always the same.

Fig. (5) shows the typical LFSR form.

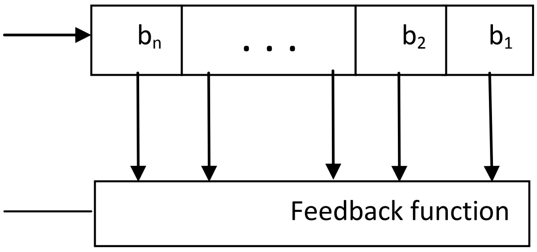


Fig. (5): Typical Linear Sequence Generator Using LFSR. [9]

5. PROPOSED LSB STEGANOGRAPHY SYSTEM

In the proposed system, details of the system lies into two phases at the sender side and also two phases at the receiver side.

a. At sender side:
   1. Encrypt the secret message using AES cryptographic algorithm.
   2. Hide the encrypted message randomly inside 2D gray scale image.

b. At receiver side:
   1. Extract the encrypted message from the stego-image.
   2. Decrypt the message using AES cryptographic algorithm, then the secret message is obtained.

5.1 Cryptographic System

AES cryptographic algorithm is used to encrypt/decrypt the secret message before hide it inside the cover image. The crypto model in the proposed system is as shown in Fig. (6) below.

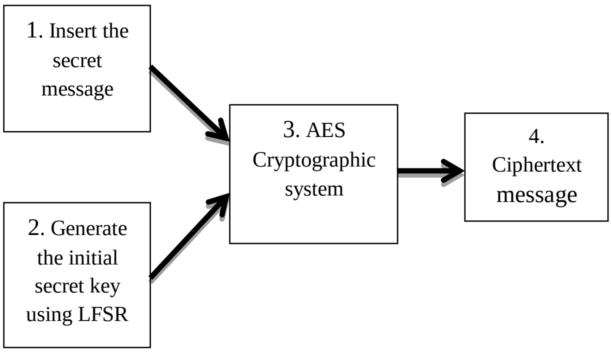

1. Insert the secret message
2. Generate the initial secret key using LFSR
3. AES Cryptographic system
4. Ciphertext message

Fig. (6): Crypto Module.

The secret message: is any secret plaintext that should be hidden in a cover image.

1. Initial secret key: it performs the initial key which is used to generate the 128 bits for the cryptography

process. The initial 16 bytes generated using LFSR with 5 bits chosen randomly from the date bits (i.e. date of day, month, and year date). Fig. (7) shows the algorithm structure. In each round, the decimal form of the five bits performs a random number.

2. AES cryptographic system: The steps involved in performing AES are as follows: AES has three approved key length: 128 bits, 192 bits, and 256 bits. This algorithm starts with a random number, in which the key and data is encrypted, which are then scrambled though four rounds of mathematical processes [8]. The key that is used to encrypt the message must also be used to decrypt it as shown in the Fig. (8).

The four rounds are called: [9]

I. **Sub Bytes**: In this we rearrange the bytes of by using a lookup table which determines what each byte is replaced with.

II. **Shift Rows**: The first row is left unchanged where as every other row is shifted cyclically by a particular offset, while. Each byte of the second row is shifted to the left, by an offset of one, bytes in the third row are shifted by an offset of two, and the fourth row by an offset of three. This is applied to all three key lengths, though there is a variance for the 256-bit block where the first row is unchanged, the second row offset by one, the third by three, and the fourth by four.

III. **Mix Columns:** a mixing operation using an invertible linear transformation in order to combine the four bytes in each column. The four bytes are taken as input and generated as output.

IV. **Add Round Key**: a round key is derived from Rijndael's key schedule, and round key is added to each byte of the row. Each round key gets added by combining each byte of the row with the corresponding byte from the round key.

3. ciphertext message: performs the encrypted secret message.

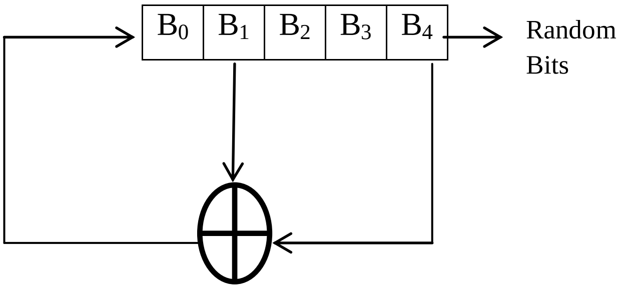


Fig. (7): LFSR with 5 bits.

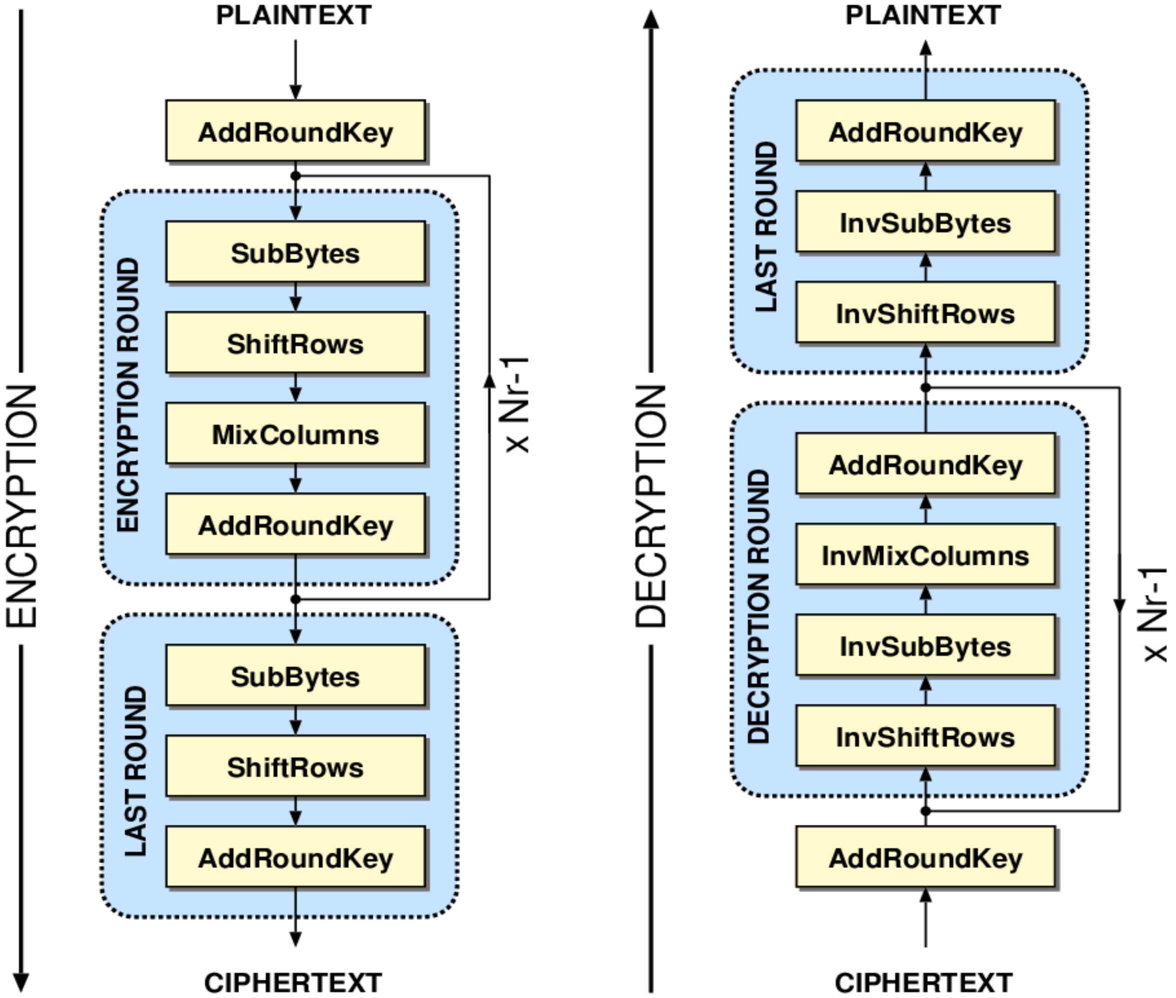


Fig. (8): AES Algorithm. [9]

5.2 Steganography System

Proposed steganography system is a LSB technique. A 2D gray scale image is used as cover file. The size of the cover image is 256x256 pixels which mean that a 65536 hiding locations are provide.

Proposed steganographic system lies onto three phases: ***generate random hiding locations***, ***permutation of the secret encrypted message bits***, and the ***hiding process***.

1. Random Hiding Locations generator:

   The first steganographic phase is how to choose the pixels for hiding? In the proposed system a random number generator based on LFSR is used.

   16 bits LFSR algorithm is used, in order to generate 256x256 pixels (i.e. 65536 pixels). Fig. (9) shows the pixels generator algorithm.

Example:

After converts the 2D image pixels to 1D pixels:

- Lets the initial 16 bits taken from the date.
- Lets the date be (D/M/Y) 11/12/2014, the binary presentation of the date is: 01011/ 1100/ 11111011110.
- Lets the initial 16 bits chosen as follows:

[y(5) y(6) y(7) y(8) y(9) y(10) y(11) m(1) m(2) m(3) m(4) d(1) d(2) d(3) d(4) d(5)]= 1011110110001011

∴ The initial state is 1011110110001011

- After applying the algorithm, the generated random numbers (ex. 25 random numbers) are:

[48548 24274 12137 38836 19418
9709 4854 2427 1213 33374
16687 8343 4171 34853 17426
41481 20740 43138 21569 10784
5392 2696 34116 17058 41297]

- So, the first pixel used for hiding is the pixel number 48548.

2. Secret encrypted message permutation:

   After converting the encrypted secret message to its equivalent binary bits, each character performs with 8 bits; each 4 random bits are hidden in the least significant 4 bits of the pixel value.

   Example:

- Lets the plaintext of the secret message is (Hi).
- Hi= [72 105] in decimal= [01001000 01101001] in binary.

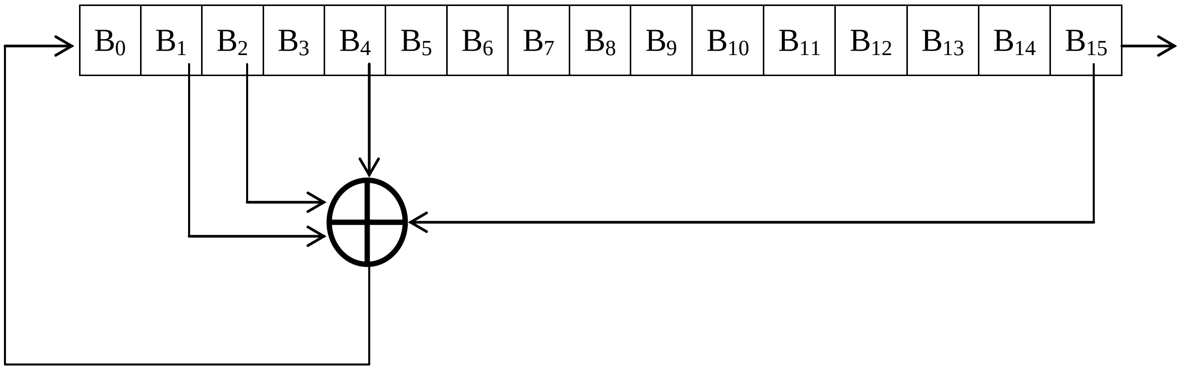


Fig. (9): 16 bits LFSR random number generator.

- The binary bits are permutated randomly using random number generator based on LFSR algorithm.
- The permutation matrix generated for this example is:

  [6 3 9 4 2 1 8 12 14 15 7 11 5 10 13 16]

  So, the first bit of the permutated bits will be the sixth one of the original binary bits.

  Permutated bits= 0000100000011101

- The random pixels generated randomly are:

  48527 57031 61283 30641

- Image pixels value of the above locations are:

  Pixels(48527)=166=10100110

  Pixels(57031)=16=00010000

  Pixels(61283)=204=11001100

  Pixels(30641)=101=01100101

3. Hiding Process:

- Now, the hiding process is done.
- Each random 4 bits of the secret encrypted message are xored with the least significant 4 bits of the pixels value.
- Continued with the example above (in section 2):

  Pixels binary values are:

  Pixels(48527)=166=10100110

  Pixels(57031)=16=00010000

  Pixels(61283)=204=11001100

  Pixels(30641)=101=01100101

  Before hiding processing, the pixels anding with 11110000

  Pixels(48527)=166=10100110 AND 11110000=10100000

Pixels(57031)=16=00010000 AND 11110000=0001**0000**

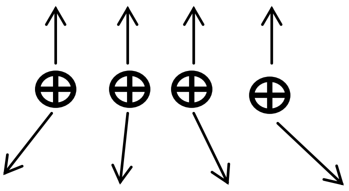
Pixels(61283)=204=11001100 AND 11110000=1100**0000**

Pixels(30641)=101=01100101 AND 11110000=0110**0000**

And the binary permutated message bits was: **0000100000011101**

Now, the xor operation is performed:

0000 **1000** **0001** **1101**



10100000 00010000 11000000 01100000

The Result = 01010000 11101000 00110001 10011101

- The new generated pixels are:

  Pixels(48527)= 01010000=80

  Pixels(57031)= 1110**1000**=232

  Pixels(61283)= 0011**0001**=49

  Pixels(30641)= 1001**1101**=157

5.3 Steganalysis

Steganalysis is the task of finding hidden messages in various media. [1]

The steganalysis system performs by the following steps:

- Extract the stego pixels (i.e. the pixels contains the secret message bits) by generate the same random location using the same algorithm described above in the steganography system.
- Extract the least significant bits of the pixels by anding the bits with 00001111.

  Example: continued from the result obtained above:

Pixels(48527)= 80=01010000 AND 00001111=00000000

Pixels(57031)=232= 1110**1000** AND 00001111=0000**1000**

Pixels(61283)= 49=0011**0001** AND 00001111=0000**0001**

Pixels(30641)= 157=1001**1101** AND 00001111=0000**1101**

- The message obtained is: **0000100000011101**
- Re permutate the message bits: **0100100001101001**
- The two message bytes are: **01001000 01101001=72 105**
- The characters of the message are: Hi

  Note: all the calculations are performs using Matlab programming language.

6. RESULTS AND ANALYSIS

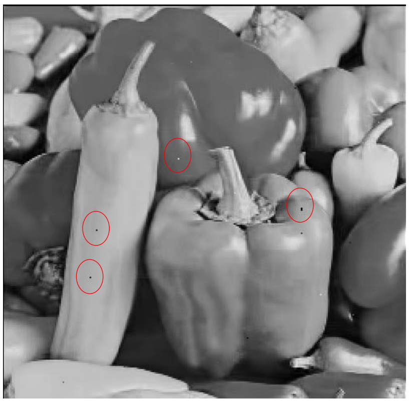
The results of the proposed system are measured with 4 bits LSB and 8 bits LSB steganography technique.

In 4 bits LSB stego system, the stego image is look to like the original message whatever the message length was. But in 8 bits LSB stego system, the stego image contains many splotch at the stego pixels as shown in the following figures below. Note the splotch on the picture in (b), the splotches are surround with red circle in order to make them clear to the viewer.

When message length increased the splotches number increased and due to the large number of splotches on the stego image, the third party can be ensure of the message existence as shown in Fig. (11) above.

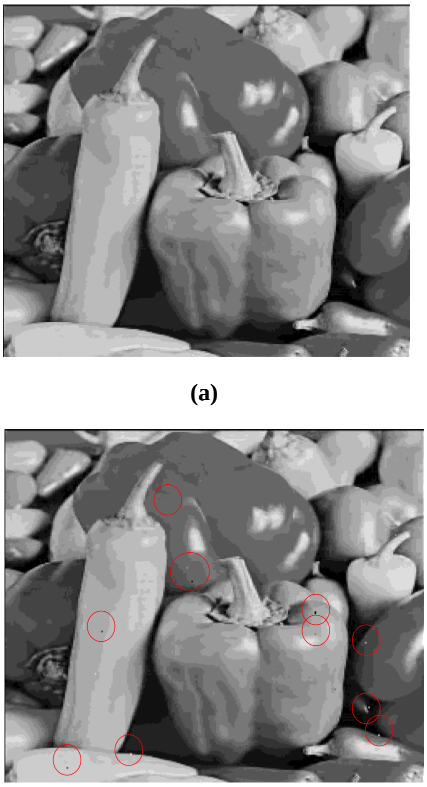
The suspension of third party of the message existence being assurance when the message lengths increase due to the increasing of splotch number when using 8 bits LSB steganography technique as shown in Fig. (12 b) and Fig. (13 b) while the stego image of 4 LSB steganography technique still like the original cover image as shown in Fig. (12 a) and Fig. (13 a).

(a)



(b)

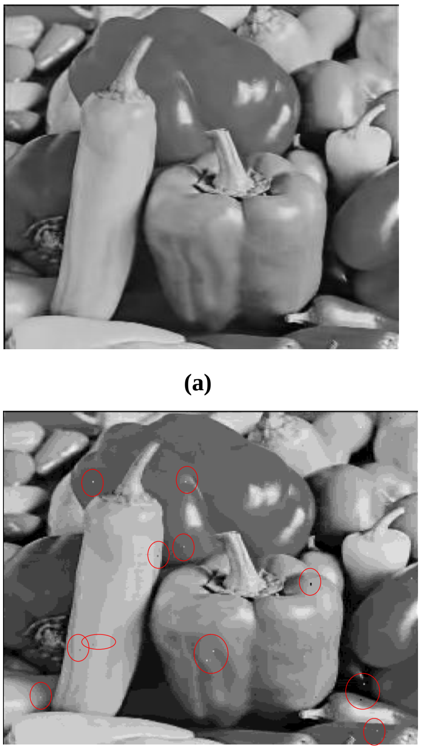
Fig. (10): (a): 4 bits LSB steganography with 8 character message length, (b): 8 bits LSB steganography with 8 character message length.



(a)

(b)

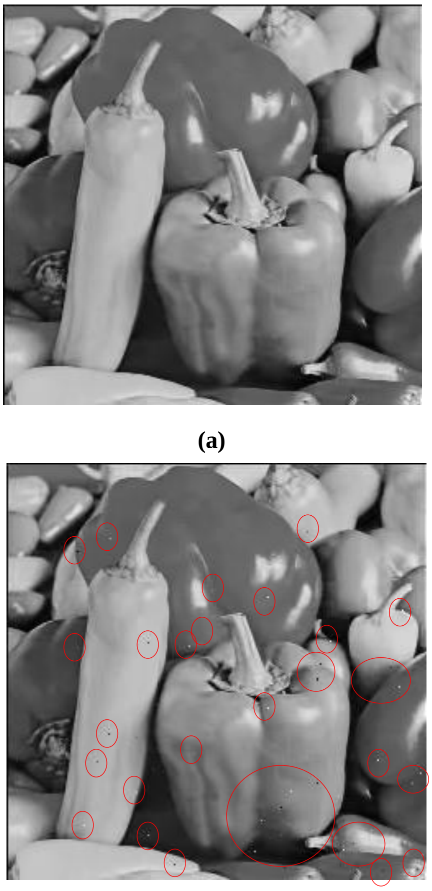
Fig. (11): (a): 4 bits LSB steganography with 16 character message length, (b): 8 bits LSB steganography with 16 character message length.



(a)

(b)

Fig. (12): (a): 4 bits LSB steganography with 32 character message length, (b): 8 bits LSB steganography with 32 character message length.



(a)

(b)

Fig. (13): (a): 4 bits LSB steganography with 64 character message length, (b): 8 bits LSB steganography with 64 character message length.

7. CONCLUSIONS

A method of embedding text-based data into a gray scale image file using the method of LSB steganography method has been presented in this paper. More security features are adding to the proposed system in this paper, by adding the randomness to the cryptography key generation algorithm, pixels chosen for the hiding process, and the randomness used to permutate the secret encrypted message.

The secret message is encrypted with truly random secret key, then the encrypted message bits permutate with truly random permutation matrix generated using PRNG which generate a random permutation matrix according to the length of the secret message; after that, the hiding process using LSB algorithm is performed.

Using 4 bits LSB is the perfect technique since, the stego image is exactly looks like the original cover image thus, the third party have no suspensions about the existence of the secret message.

Using 4 LSB steganographic technique, a message with length 327680 character can be hidden inside the gray scale image.

8. REFFRENCES

[1] Ashish kumari, Shyama Sharma, and Navdeep Bohra, "Implementation of IMAGE STEGANOGRAPHY Based on Random LSB", International Journal of Computer Science and Management Studies (IJCSMS),

Vol. 12, Issue 01, 2012.

[2] Mamta Juneja and Parvinder Singh Sandhu, " Information Hiding using Improved LSB Steganography and Feature Detection Technique" , International Journal of Engineering and Advanced Technology (IJEAT), Volume-2, Issue-4, ISSN: 2249 – 8958, 2013.

[3] Dr.Shubhangi D.C and Manikamma Malipatil, "Authontication Watermarking for Transmission of Hidden Data Using Biometrics Technique", International Journal of Emerging Technology and Advanced Engineering, Volume 2, Issue 5, ISSN 2250-2459, 2012.

[4] ELTYEB E. ABED ELGABAR and FAKHRELDEEN A. MOHAMMED, "JPEG versus GIF Images in forms of LSB Steganography", International Journal of Computer Science and Network (IJCSN), Volume 2, Issue 6, 2013.

[5] Atallah M. Al-Shatnawi, "A New Method in Image Steganography with Improved Image Quality", Applied Mathematical Sciences, Vol. 6, no. 79, 3907 – 3915, 2012.

[6] Babita and Mrs. Ayushi, "Secure Image Steganography Algorithm using RGB Image Format and Encryption Technique", International Journal of Computer Science & Engineering Technology (IJCSET), Vol. 4, No. 06, ISSN : 2229-3345, 2013.

[7] K. Chandra Sekhar, M.Chandra Sekhar, and Mr. K.Chokkanathan, " teganography: A Security Model for Open Communication", Int. J. Advanced Networking and Applications, Volume: 04, Issue: 04, Pages:1690-1694, ISSN : 0975-0290, 2013.

[8] Henk C. A. van Tilborg (Ed.), "Encyclopedia of cryptography and security", pp.159, Springer, 2005.

[9] Jyotika Kapur and Akshay. J. Baregar, "Security using image processing", International Journal of Managing Information Technology (IJMIT) Vol.5, No.2, 2013.

[10] Dr. Ashish Negi et al. ," Cryptography Playfair Cipher using Linear Feedback Shift Register", IOSR Journal of Engineering, Vol. 2(5) pp: 1212-1216, ISSN: 2250-3021, 2012.