Detecting the LSB-LFSR Algorithm Stego-Image Using the Statistical Analysis and Visual Attack

Dr. Ayad G. Naser — **Ministry of Education**

Faez H. Ali — **Al-Mustansiriya University**

Abstract

The aim of this paper is to detect the hidden information in BMP images when using Least Significant Bit (LSB) technique. In general, the passive steganalysis system deals with some stego-tools based on three stages: Diagnosis, Breaking and Extraction. In this paper we focus in Diagnosis stage to attack a Linear Feedback Shift Register (LFSR) algorithm as stego tools. In the diagnosis process, the statistical analysis and visual test are suggested as tools for detection. These tests are: Mean Square Error (MSE) for stego-cover images, and Laplace Operator, chi-square analysis and visual test for stego only attack.

1. Introduction

The simplest way of hiding information in a sequence of binary numbers is replacing the Least Significant Bit (LSB) of every element with one bit of the secret message. Since flipping the LSB of a byte only means the addition of a small quantity, the sender assumes that the difference will lie within the noise range and that will therefore not be generally noticed. This algorithm changes the statistical proprieties of the cover significantly, even if the message consists of truly random bits. The LSB technique can be improved by using a pseudo random generator.

Simply embedding information into the LSB of an image provides no protection if the scheme produces artifacts. For this reason, the research is chosen to warn the users who are using hiding tools which depend on LSB techniques.

Fridrich [1,2] introduced the dual statistics steganalytic method for detection of LSB embedding in uncompressed formats. For high quality images taken with a digital camera or a scanner, the dual statistics steganalysis indicates that the safe bit rate is less than 0.005 bits per sample, providing a surprisingly stringent upper bound on steganographic capacity of simple LSB embedding.

Pfitzmann and Westfeld [3] introduced a method based on statistical analysis of Pairs of Values (PoVs) that are exchanged during message embedding. Pairs of Values that differ in the LSB only, for example, could form these PoVs. This method provides very reliable results when we know the message placement (such as sequential).

We can classify all the introduced steganalysis tools into two main methods; the first is the visual analysis which is consist of visual test and Laplace operator, while the second method is the statistical analysis which is consists of mean square error tool and chi-square test.

The hiding algorithm consists of single LFSR of LSB BMP images hiding systems are suggested in order to be detected. A new proposed steganalysis system is constructed to detect the LSB BMP images.

2. Hiding in Images

In this section we deal with data encoding in still digital images. In essence, image steganography is about exploiting the limited powers of the human visual system (HVS) [5]. Within reason, any plaintext, cipher text, other images, or anything that can be embedded in a bit stream can be hidden in an image. Image steganography has come quite far in recent years with the development of fast, powerful graphical computers, and steganographic software is now readily available over the Internet for everyday users.

Information can be hidden in many different ways in images. Straight message insertion can be done, which will simply encode every bit of information in the image. More complex encoding can be done to embed the message only in "noisy" areas of the image that will attract less attention. The message may also be scattered randomly throughout the cover image [6].

The most common approaches to information hiding in images are:

1. **Least significant bit (LSB) insertion.**
2. **Masking and filtering techniques.**

The least significant bit insertion method is probably the most well known image steganography technique. It is a common, sample approach to embedding information in a graphical image file. Unfortunately, it is extremely vulnerable to attacks, such as image manipulation.

When LSB techniques are applied to each byte of a 24-bit image, three bits can be encoded into each pixel. Any changes in the pixel bits will be indiscernible to the human eye. For example, the letter A can be hidden in three pixels. Assume the original three pixels are represented by the three 24-bit words below:

0	0	1	0	0	1	1	1	1	1	1	0	1	0	0	1	1	1	0	0	1	0	0	0
0	0	1	0	0	1	1	1	1	1	0	0	1	0	0	0	1	1	1	0	1	0	0	1
1	1	0	0	1	0	0	0	0	0	1	0	0	1	1	1	1	1	1	0	1	0	0	1

The binary value for the letter A is.

0	1	0	0	0	0	0	1

Inserting the binary value of A into the three pixels, starting from the top left byte, would result in:

0	0	1	0	0	1	1	1	1	1	1	0	1	0	0	0	1	1	0	0	1	0	0	0
0	0	1	0	0	1	1	0	1	1	0	0	1	0	0	0	1	1	1	0	1	0	0	0
1	1	0	0	1	0	0	1	0	0	1	0	0	1	1	0	1	1	1	0	1	0	0	1

The emphasized bits are the only bits that actually changed. The main advantage of LSB insertion is that information can be hidden in the least and second two least bits and still the human eye would be unable to notice it [7].

3. Steganalysis [9]

Steganalysis is the art of discovering hidden data in cover objects. As in cryptanalysis, we assume that the steganographic method is publicly known with the exception of a secret key. The method is secure if the stego-images do not contain any detectable artifacts due to message embedding. In other words, the set of stego-images should have the same statistical properties as the set of cover-images. The ability to detect secret messages in images is related to the message length. Obviously, the less information we embed into the cover-image, the smaller the probability of introducing detectable artifacts by the embedding process.

In general, steganalysis is carried out for breaking the security of a steganographic system .It is assumed that an adversary has the knowledge of the system and has one or many images are intercepted from a public channel. Security of the system lies solely on the secrecy of the key. The modules required for steganalysis are shown in figure (2) [10].

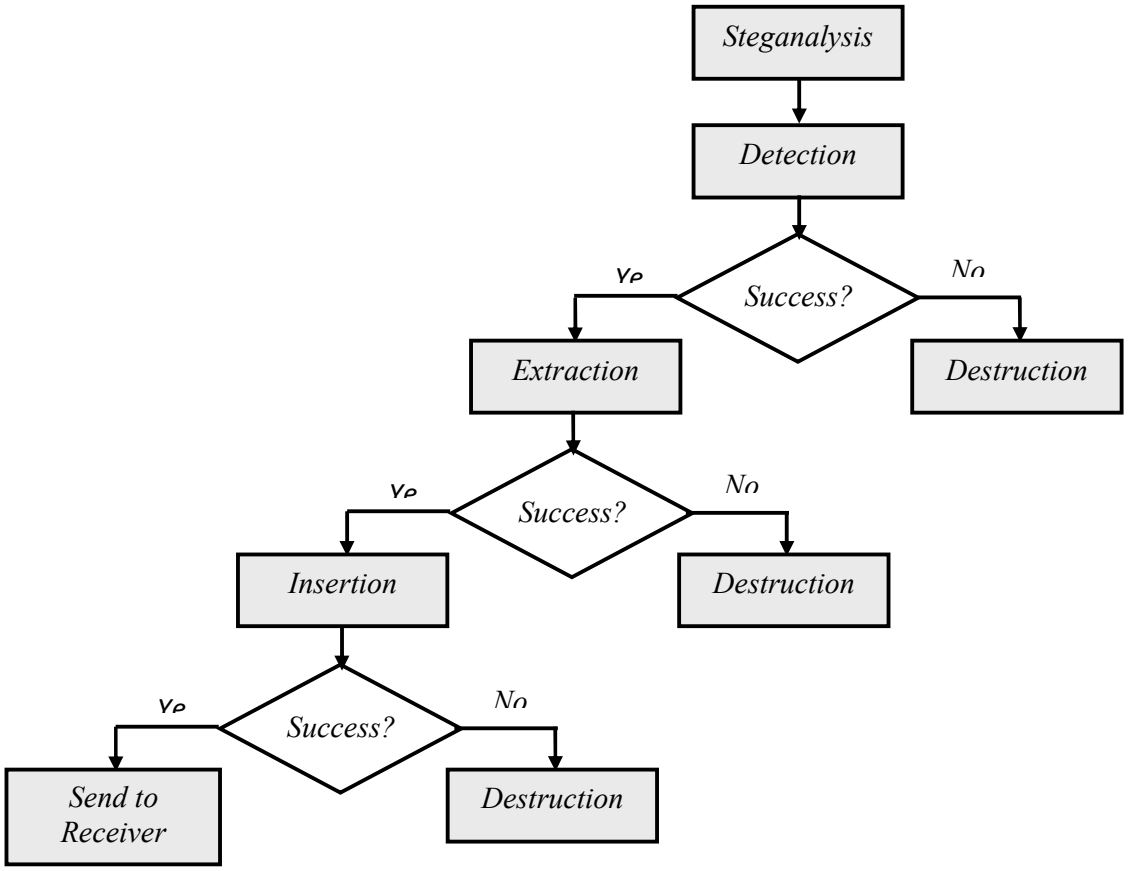


Figure (2) Basic modules in steganalysis.

3.1 Detecting Hidden Information [12]

The image format and its size can also provide clues about the type of stego-tools used .The next step is to identify the signatures (if available) of specific tools in the image. Under normal circumstances, it is difficult to obtain the secret message or the host image.

3.2 Extracting [13]

Once steganographic contents are detected with high probability, the second step is of message extraction. In many countries it may be possible to obtain legal permission for getting the password required to extract the actual hidden information. Otherwise, the steganalyst may have to use a good amount of infrastructure to carry out a brute force or dictionary attack.

3.3 Disabling and Modification Steganography [12]

The disabling or removal of hidden information in images comes down to image processing techniques. For LSB method of inserting data, simply using lousy compression techniques, such as JPEG, is enough to render the embedded message useless. Images compressed with such a method are still pleasing to the human eye but no longer contain the hidden information.

4. Steganalytic Methods [10]

With careful selection of an appropriate cover image and stego-tool it is possible to create a stego-image that does not appear to be different within the limits of human perception. However, electronically each of these tools leaves a fingerprint or signature in the image that can be used to alert an observer to the presence of hidden message.

4.1 Laplace Operator [11]

The passive attacker can detect the existence of a secrete message by using different ways, the most common method is the discrete Laplace operator. By this operator it is possible to detect secret message in grayscale images:

$$\nabla^2 p(x,y)=p(x+1,y)+p(x-1,y)+p(x,y+1)+p(x,y-1)-4p(x,y). \quad \ldots(1)$$

The value of the point (x,y) in equation (1) gives the "Laplace filtered" image. Since we can expect neighboring pixels to have a similar color, the histogram of Laplace filtered is tightly clustered around zero, which is shown in figure (1).

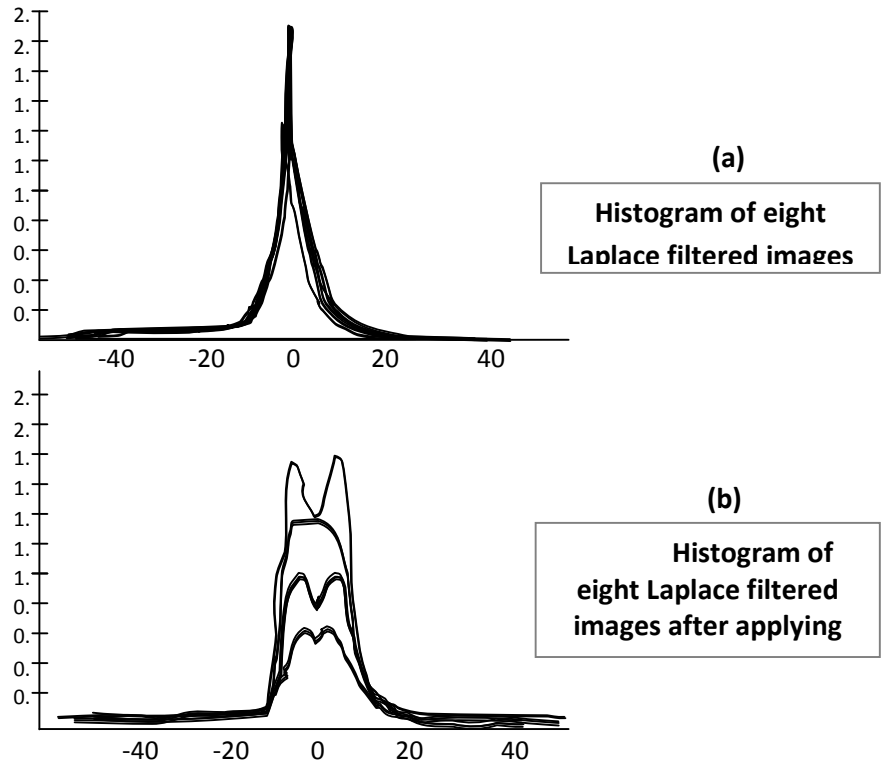


Figure (1) Histogram of Laplace filtered

Figure (1-a) shows eight histograms of Laplace filtered grayscale images printed in one coordinate system. Figure (1-b) shows the histogram of the same image after applying an existing steganography algorithm. Since the embedding process adds noise to the picture, which is statistically quite different from the true random noise, the new histogram differs extremely. Laplace filtering does not prove the existence of a secret, but it will provide strong evidence that the picture was subject to modification.

4.2 Visual Test

The visual attack is a stego-only attack that exploits the assumption of most authors of steganography programs that the least significant bits of a cover file are random. Relying on a human to judge if an image presented by a filtering algorithm contains hidden data, or does not. The filtering algorithm removes the parts of the image that are covering the message. The output of the filtering algorithm is an image that consists only of the bits that potentially could have been used to embed data. The filtering of the potential stego image is dependent on the steganographic embedding function that is analyzed. However, as most of the embedding functions are similar in most cases only small changes are necessary to adapt an existing filtering algorithm to another steganographic embedding function.

4.3 Known Cover Attack Using Mean Squared Error [10]

To calculate the Mean Squared Error (MSE) between the original image and suspected image, we must know the difference of pixel color in the both images. The result will be the square amount of errors depending on the size of these images.

The equation that is used to calculate (MSE) is:

$$MSE=\frac{I}{XY}\sum_{x,y}\left(O_{x,y}-\hat{O}_{x,y}\right)^2 \qquad \text{...(2)}$$

X = number of rows

Y = number of columns

$O_{x,y}$ = value of pixel in the position x,y of cover-image

$\hat{O}_{x,y}$ = value of pixel in the position x,y of stego-image

4.4 Stego-Only Attack Using Chi-Square Test [7]

We now look at two adjacent color values (a Pair of Values, also referred to as PoV), where adjacent means identical except for the least significant bit: When overwriting the least significant bits of all occurrences of one of these color values with a bit from the secret message, the frequencies of these two color values will essentially be the same. This happens because the data that is embedded is encrypted and therefore equally distributed.

The idea of the statistical attack is to compare the frequency distribution of the colors of a potential stego file with the theoretically expected frequency distribution for a stego file. The theoretically expected frequency distribution is calculated as follows: Under the assumption that only the least significant bits are overwritten and that the embedded data is equally distributed the expected frequency distribution is that for each PoV the frequencies of the two colors are the same. Due to the fact that the sum of the occurrences of the two colors in a PoV is not changed by the embedding process, the expected frequency can be calculated as the median of the frequencies of a PoV in the potential stego file.

The degree of similarity of the frequencies in the potential stego file and the theoretically expected frequencies is a measure for the probability that the analyzed file contains a hidden message. Statistical tests can reveal if an image has been modified by steganography by testing whether an image's statistical properties deviate from a norm [9].

The Chi-square test is used to determine whether color frequency distribution in an image shows distortion from embedding hidden data. Because the test uses only the stego medium, the expected distribution y_i^* for the χ^2-test has to be computed from the image. Let n_{2i} be the frequency of two adjacent color values in the image. We assume that an image with hidden data embedded has similar frequency for two adjacent color values. As a result, we can take the arithmetic mean:

$$y_i^* = \frac{n_{2i} + n_{2i+1}}{2} \qquad \text{...(3)}$$

to determine the expected distribution. The expected distribution is compared with the observed distribution:

$$y_i = n_{21} \qquad \text{...(4)}$$

the value for the difference between the distributions is given as:

$$\chi^2 = \sum_{i=1}^{v+1} \frac{(y_i - y_i^*)^2}{y_i^*} \qquad \text{...(5)}$$

where ν is the degrees of freedom, that is, the number of different categories in the histogram minus one. The probability p that the two distributions are equal is given by the complement of the cumulative distribution function,

$$p = 1 - \int_0^{\chi^2} \frac{t^{(v-2)/2} e^{-t/2}}{2^{v/2}\Gamma(v/2)}\, dt \qquad \text{...(6)}$$

For example the Chi-square test can be explained as follows:

Objective: is there a connection between two categorical variables?

1) H_0 : Variable 1 is independent of Variable 2 (no relationship).

2) H_A : Variable 1 is related to Variable 2.

3) Choose significance level α (from literature survey typically, α=0.05).

We can compute the probability of embedding for different parts of an image. The selection depends on what steganographic system we try to detect. For an image that does not contain any hidden data, probability of embedding to be zero everywhere [14].

5. Steganography System

In this paper, a single LFSR-steganography system introduced to hide information in BMP images, using LSB technique. Of course, the system tests the size of the message want to be hidden before hiding process is started and compared with container image size to be sure that the container can contain

the data of the message. Figure(2) shows the block diagram of steganography system.

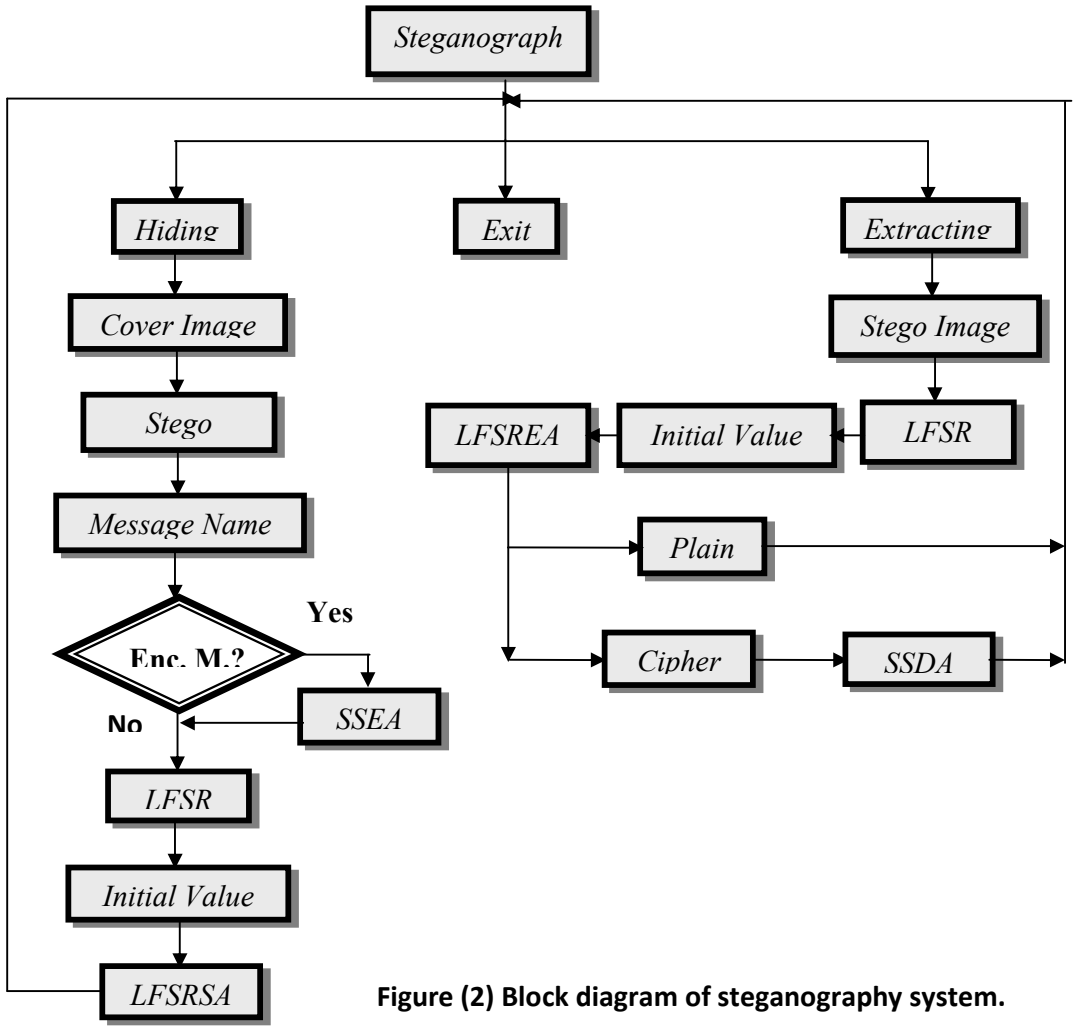


Figure (2) Block diagram of steganography system.

It is important to mention that the message data should be converted to binary to be more suitable for hiding process. Converting process can use any 5-binary encoding system.

The pseudo code of the character to binary converting algorithm is:

```
NAME   : Character to Binary Converting Algorithm (C2BCA).
INPUT  : Message data;
PROCESS        : Repeat
                    Read message character(i);            {i=1..Message length}
                    bit(i,j) = character(i)[bit(4),bit(3),bit(2),bit(1),bit(0)];  {j=1..5}
```

The pseudo code of the binary to character converting algorithm is:

```
NAME   : Binary to Character Converting Algorithm (B2CCA).
INPUT  : Message binary data;
PROCESS        : i = 0, j = 0;
                  Repeat
                    Read message bit(j);
                    j =  j+1;
                    if j mod 5 = 0 then
```

The message could be plain or cipher text. The simple substitution is chosen as an example of an encipher systems, which the alphabet is scrambled, and each plain text letter maps to a unique cipher text letter. A permutation is a reordering of the elements of a series. The pseudo code of the simple substitution encipher algorithm is:

```
NAME   : Simple Substitution Encipher Algorithm (SSEA).
INPUT  : Plain Message;
PROCESS        : Repeat
                    Read plain character(i);        {i=1..Message length}
                    cipher character(i) = encipher table [plain character(i)];
```

The pseudo code of the simple substitution decipher algorithm is:

```
NAME   : Simple Substitution Decipher Algorithm (SSDA).
INPUT  : Cipher Message;
PROCESS        : Repeat
                    Read cipher character(i);       {i=1..Message length}
                    plain character(i) = decipher table [cipher character(i)];
```

Let us suppose that the start hiding byte is byte number 1000.

This system depends on a simple algorithm represented by Linear Feedback Shift Register (LFSR) with 11-stages length. This algorithm uses a stego key called Basic Key (BK), 10-bit length used as an initial value to LFSR and the last stage fills by (1), the BK must be known to both transmitter and receiver, and must be changed every period of time (weekly, daily or every message). When the LFSR is filled with initial values, it starts to move to generate pseudo random value which represents a key jump (Keyjmp) to specify the next hiding position and so on. The diagram of this steganography system is showed in figure (3).

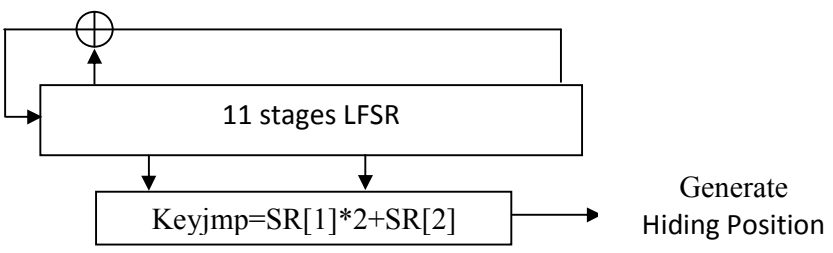


Figure (3) LFSR Steganography system.

The pseudo code of the LFSR steganography algorithm is:

```
NAME   : LFSR Steganography Algorithm (LFSRSA).
INPUT  : Cover image, plain message,
                BK (10 bits) as initial values to LFSR;
PROCESS       : If want to hide encipher message CALL SSEA;
                CALL C2BCA;
                i = 1000, j = 1;
                Repeat
                   Read Cbyte(i);
                   Read bit(j);
```

While the pseudo code of the LFSR extracting algorithm is:

```
NAME   : LFSR Extracting Algorithm (LFSREA).
INPUT  : Stego image,
                BK (10 bits) as initial values to LFSR;
PROCESS       : i = 1000, j = 1;
                Repeat
                   Read Sbyte(i);
                   extract bit(j) from Sbyte(i);
                   Write bit(j) in message;
```

6. Detecting System Design

Any steganalysis system could be divided into three main stages, these stages are hidden message diagnosis, breaking and extracting. Every stage implementation is related to implementation success of the previous stage. The implementation of every stage is related to how much the available information is useful .In this paper, the origin image availability and some information about the stego tools are only the available information, and there is no information about the hidden message, like message length or probable words.

One stage may be implemented more than one style or method to guarantee the stage implementation correctly and precisely, that is because one style may not be sufficient to successfull implementation on the specified stage. We will focus in detecting stage only which it's represented by Hidden Message Diagnosis Stage (HMDS). The HMDS is important, since it's the first stage toward the breaking stage. It saves time, in searching for the hidden message or diagnoses the stego tools. The diagnosis stage may not succeed in specifying the existence of the hidden message; therefore, it must apply more than one diagnosis style to gain a precise decision. Of course, the successful diagnosis stage implementation is related to the available information. Figure (4) shows the block diagram of steganolysis system.

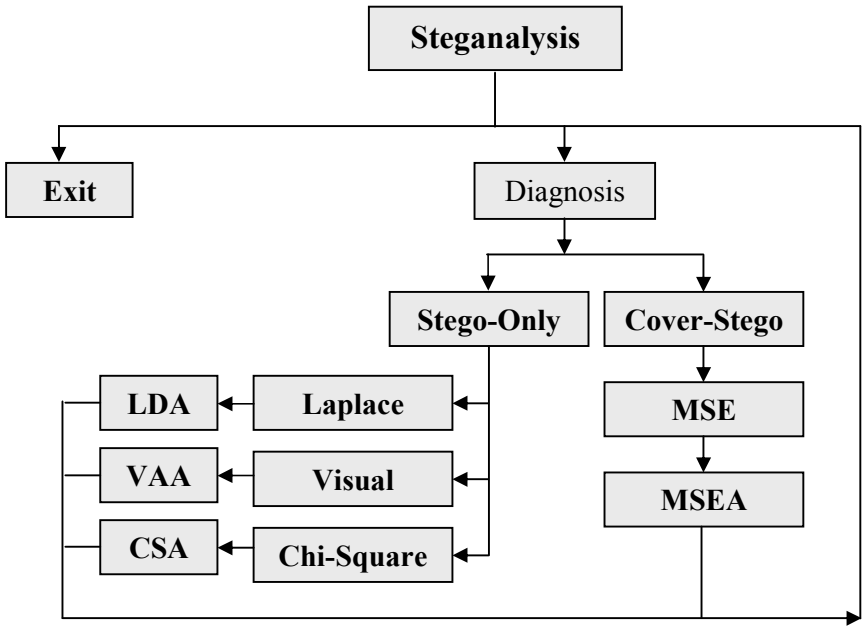


Figure (4) Block diagram of steganolysis system

6.1 Diagnosis by Cover and Stego Images

In this method, the cover and the stego images are compared, that is done by using MSE. To calculate the MSE, equation (1) can be used in bytes of the two images to determine whether the second image contains a hidden message, if MSE = 0, this means the second image does not contain any embedded message, else it does. As MSE is high, this is a function to the size of the hidden message. The type of stego tool has no real effect on MSE value. Table (1) shows different cases of hiding and results of MSE.

Table (1) MSE results.

Stego tool	Message	MSE	
		1000 bits	2000 bits
Without hiding	No message	0.00000	0.00000
LFSRSA	Plain	0.02049	0.041213
	Cipher	0.02467	0.049242

The pseudo code of the calculating MSE Algorithm is:

```
NAME   : MSE Algorithm (MSEA).
INPUT  : Cover image, Stego image;
PROCESS        : Calculate X, Calculate Y
                 i = 0..X-1, j = 0..Y-1,MSE = 0;
                 Repeat
                   Read Cbyte(i,j);
```

6.2 Diagnosis by Stego Image Only

Diagnosis by stego image only is more difficult than diagnosis by cover and stego, and sometimes, no final decision can be made as it could be in diagnosis by cover and stego. Therefore, three methods can be applied to help the steganalyst in getting precise decision to move to the next stage of image analysis or abort the steganalysis.

The three diagnosis methods which are mentioned before, visual attack using filter, statistical attack using Laplace operator and Chi-square are be applied in this subsection.

6.2.1 Visual Attack Method

In this method, a filter can be used through calculating the LSB of image bytes then multiplying by scale value, if the image contains hidden message we expect some noise to appear in the filtered image. This method depends on showing the filtered image so it could be useless in some Bmp images.

The pseudo code of the visual attack algorithm is:

```
NAME  : Visual Attack Algorithm (VAA).
INPUT : Stego image, Scale;
PROCESS      : Calculate X, Calculate Y;
               i = 0..X-1, j = 0..Y-1;
               Repeat
               Read Sbyte(i,j);
```

6.2.2 Laplace Operator Method

Laplace operator is useful in checking if there is a deviation in neighborhood pixels, as usual the neighborhood pixels are approximate to each others. Equation (1) can be used to make frequency to this operator and then graph the result about some range of Laplace operator possible values.

The pseudo code of the Laplace operator algorithm is:

```
NAME  : Laplace Operator Algorithm (LOA).
INPUT : Stego image;
PROCESS      : Calculate X, Calculate Y;
               i = 1..X-2, j = 1..Y-2;
               Repeat
                  Read P(i,j), P(i+1,j), P(i-1,j), P(i,j+1), P(i,j-1) from stego;
```

6.2.3 Chi-Square Method

This method is represented by calculating PoV, first, the color frequency must be calculated in every percentage in increasing form (1%, 2%,…, 100%), of course the occurrences of the colors and their frequencies will increase, that means increasing in the number of different categories (degree of freedom).

Equation (4) can be applied to calculate the Chi-square value with degree of freedom ν, depending on equation (5), the hiding probability p can be calculated, if $p \geq 0.05$ there is a good probability of hiding, in contrast, the image may contain no hidden message. We can use the results to graph a histogram to describe the probability of hiding, and may specify the probable length of the hidden message.

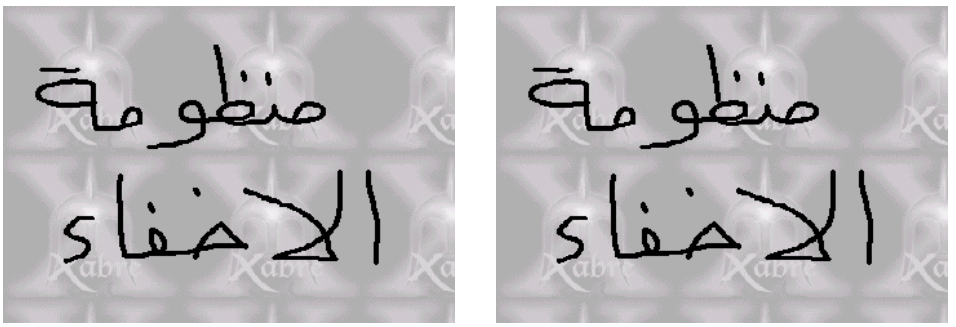
The pseudo code of the Chi-Square algorithm is:

```
NAME   : Chi-Square Algorithm (CSA).
INPUT  : Stego image;
PROCESS    :  i = 0..X-1, j = 0..Y-1, k = 0,L=0, Per = (X*Y) Div 100;
              Repeat
                 Read P(i,j) from stego;
                 FREQ=FREQ[P(i,j)]+1;
                 If L mod Per = 0 then;
```

7. Experimental Examples

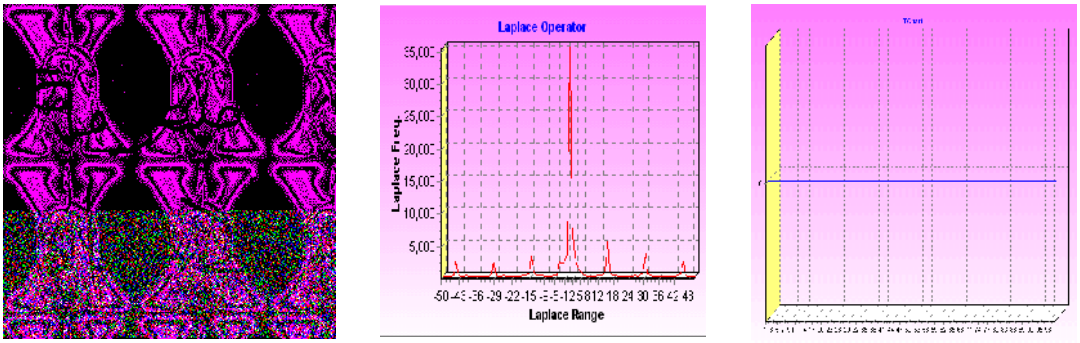
In this section, two images are introduced; to hide one message (plain and cipher) use the steganography system. The detecting represented by diagnosis stage results are shown using chi-square test graph.

Figure (4) shows image before and after hiding process using LFSRSA. The diagnosis stage results of image showed in Figure (5).



(a) **(b)**

Figure (4) shows image (a) before and (b) after hiding process using LFSRSA.



(a) (b) (c)

Figure (5) Image after applying (a) visual (b) Laplace (c) Chi-Square tests.

8. Steganalysis System Efficiency Tests

The steganalysis system uses (14) images to tests the MSE and chi-square analysis efficiency, the tests done as follows:

1. Test of hiding for LFSR algorithm steganography tools.

2. Test the diagnosis stage for hidden cipher text only.

3. Using the MSE test for cover and stego images.

4. Using the visual, Laplace and chi-square tests for stego image only.

The tests results are describe in table (2), which is shows, the successful percentage for MSE and chi-square tests.

Table (2) MSE and chi-square tests results.

Stego tools	Successful percentage			
	MSE Test	Visual	Laplace	Chi-square
LFSRA	100 %	93%	71%	57%

9. Conclusions

The research contribution based on investigating steganography in BMP images; include the steganalysis of LSB data-hiding techniques, and attacks against hidden information. The proposed Steganalysis system suggests attacking and analyzing hidden information in LSB BMP images.

The statistical tests, visual tests and histograms are used to decide if suspected image has information hidden or not. the following are some points concluded from this study:

1. **The steganography system designer must follow some countermeasures concluded from steganalysis tools to protect his steganography systems.**
2. **In this paper, many stego-images are tested, we notice that not every stego image can be detected by using steganalysis attacks.**
3. **Not every steganalysis attack can gives a positive result gotten from stego image. This conclusion is obtained from the many tests were done in many stego-images.**
4. **The MSE value could be used as a function of the length of the hidden text.**
5. **From the results shown in table (1) and table (2), the visual and Laplace test more efficient than chi-square test, and it's not easy to diagnose the hiding using LFSR algorithm.**

10 References

[1]. Fridrich, J., Goljan, M., and Du, R., **"*Detecting LSB Steganography in Color and Gray-Scale Images***", Magazine of IEEE Multimedia, Special Issue on Security, October-November issue, 2001, pp. 22–28.

[2]. Fridrich, J., Goljan, M., and Du, R., "***Reliable Detection of LSB Steganography in Grayscale and Color Images***", Proc. ACM, Special Session on Multimedia Security and Watermarking, Ottawa, Canada, October 5, 2001, pp. 27–30.

[3]. Westfeld A., and Pfitzmann A. "***Attacks on Steganographic systems***" In proceedings of Information Hiding-Third International Workshop, vol.1768, Springer-Verlag, Berlin, 2000, pp. 61-75.

[4]. Bourke, P. "***BMP Image Format***", Englewood cliffs: Prentice-Hall, July 1998.

[5]. Umbaugh, S. E., "***Computer Vision and Image Processing: A Practical Approach Using CVIP Tools***", Prentice-Hall PTR, 1998.

[6]. Sellars, D., "***An Introduction to Steganography***", http://www.cs.uct.ac.za/courses/papers99/dsellars/stego.ps.gz, 1999.

[7]. Johnson, N. F., Duric, Z. and Jajodia, S., "***Information Hiding: Steganography and Watermarking-Attacks and Countermeasures***". Kluwer Academic Publishers, Boston Dodrecht London, 2000.

[8]. Memon Nasir D., Khalid Sayood**, "*Lossless Compression of Color Image in the RGB Domain***", Computer Science and Mathematics, Arkansas State University, 2001.

[9]. Provos, N., "***Defending Against Statistical Steganalysis***", Center for Information Technology Integration, University of Michigan, 2001.

[10]. Al-hamami. Mohammed, "***Information Hiding Attack in Image***", M. Sc. Thesis Introduced to Iraqi Commission for Computer & Informatics, Informatics Institute for Postgraduate Studies, 2002.

[11]. Lala Zareh Avedissian, "***Image in Image Steganography System***". PH. D. Thesis, University of Technology, 2000.

[12]. Curran, K. and Bailly, K., "***An Evaluation of Image Based Steganography Methods***", Internet Technologies Research Group, University of Ulster, 2001.

[13]. Hansmann, F., "***Steganalysis: Scanning the Web***", Steganos Website, URL, 2001, http://www.demcom.deutsch/index.htm.

[14]. Ahmed, G. A., "***Image Steganalysis***", M. Sc. Thesis Introduced to Informatics, Institute for Postgraduate Studies of the University of Technology, 2005.

كشف الاخفاء في الصور التي تعتمد خوارزمية المسجل الزاحف الخطي مع تقنية الثنائي الادنى باستخدام التحليل الاحصائي والمهاجمة المرئية

م.د.اياد غازي ناصر
وزارة التربية
المديرية العامة للتعليم المهني

م.فائز حسن علي
قسم الرياضيات/كلية العلوم
الجامعة المستنصرية

المستخلص

هدف البحث هو كشف المعلومات المخفية في صور BMP التي تستخدم تقنية الثنائي الادنى (LSB). بشكل عام، نظام تحليل الاخفاء غير الفعال سيتعامل مع وسائل اخفاء بالاعتماد على ثلاث مراحل: التشخيص، الكسر والاستخلاص. في هذا البحث سوف نحاول التركيز على المرحلة الاولى وهي مرحلة التشخيص لمهاجمة نظام اخفاء يعتمد خوارزمية المسجل الزاحف ذو التغذية المرتدة (LFSR). في عملية التشخيص تم اقتراح التحليل الاحصائي والاختبار المرئي كوسائل للكشف. هذه الاختبارات هي: حساب معدل الخطأ (MSE) عند توفر الصورة الاصلية وصورة الاخفاء، ومؤثر لابلاس وتحليل مربع كاي والاختبار المرئي للمهاجمة عند توفر صورة الاخفاء فقط.