DESIGN OF NEW STEGANOGRAPHY MECHANISM FOR IMAGES BASED ON LSB TECHNIQUE

M.Sc. Mohammed G. S. Al_Safi

Al_Esraa Unversity College

Abstract

In this paper, a new Least Significant Bit (LSB) nonsequential embedding technique in 24 or 8 bit colors digital Bmp images, introduced. The idea is using an arbitrarily non least bit of the chose byte to be condition of embedding bit. This technique called a Random Control Bit (RCB).

In the paper a design and implementation for images steganography system based on LSB mechanism was presented and discussed. Different image files stored by using bitmap format were utilized. Some auxiliary processes were suggested and investigated in order to recover some weak aspect inherent with the pure implementation of stego-systems. Among the auxiliary processes is the hopping and stream ciphering algorithm. Besides, the suggested system using crypto-hiding pseudo random key generator. This key generator works for the two purposes to investigate the encryption and embedding processes. The suggested RCB stego-system was tested using visual, Laplace and chi-square tests, the new method based on the idea of increasing the hoping rate due to the HVS-poor sensitivity. The results have indicating a goodd hiding performance.

المستخلص :

في هذا البحث نقدم تقنية جديدة تمثل تضمين غير متسلسل تعتمد تقنية الثنائي الاقل اهمية Least Significant Bit (LSB) في صور Bmp الرقمية الملونة من حجم 24 او 8 ثنائي. الفكرة تتضمن اختيار ثنائي اختياري من البايت المختار يختلف عن LSB ليكون شرط للثنائي المراد تضمينه. تم اطلاق تسمية هذه التقنية بتقنية الثنائي العشوائي المسيطر Random Control Bit (RCB).

نعرض في هذا البحث التصميم والتنفيذ لنظام اخفاء في الصور يعتمد آلية LSB باستخدام بعض الصور المختلفة. تم اقتراح وتحقيق بعض العمليات المساعدة لاعادة تأهيل بعض نقاط الضعف الموروثة. بالاعتماد على تلك العمليات المساعدة تم استخدام خورازميات القفز في الصورة ونظم التشفير الانسيابي، لذلك فأن النظام المقترح يستخدم مولد مفاتيح تشفير–اخفاء شبه عشوائي Crypto–Hiding Pseudo Random Key Generator. ان هذا المولد يعمل على انجاز عمليتي التشفير والتضمين معا.

لقد تم فحص نظام RCB Stego المقترح من خلال ثلاث طرق كشف وهي الاختبار البصري، اختبار لابلاس واختبار مربع كاي، ان النظام الجديد يعتمد فكرة زيادة معدل القفزة للمحافظة على عدم الحساسية لنظام الرؤية البشري وقد كانت النتائج الفحص ناجحة.

1. Introduction

Bitmap (BMP) file format is used for bitmap graphics on the window platform only. Unlike other file formats, which store image data from top to bottom and pixels in Red, Green, Blue order, the BMP format stores image data from bottom to top and pixels in Blue, Green, Red order. When saving a file in the BMP format, add the ".BMP" file extension to the end of its file name [1].

Steganography, from the Greek, means covered, or secret writing, and is a long-practiced form of hiding information. Although related to cryptography, they are not the same. Steganography's intent is to hide the existence of the message, while cryptography scrambles a message so that it cannot be understood. The **advantage** of steganography is that it can be used to secretly transmit messages without the fact of the transmission being discovered. Often, using encryption might identify the sender or receiver as somebody with something to hide [2].

Image Domain tools encompass bit-wise methods that apply least significant bit (LSB) insertion and noise manipulation. These approaches are common to steganography and are characterized as "simple systems" in [3]. The tools used in this group include two kinds applied either directly to pixel values to indices in palette images (StegoDos, S-Tools [4], Mandelsteg and EzStego [5]).

In this paper, the following aims are achieved:

1. A new pseudo random crypto-hiding key generator is constructed to generate two different pseudo random sequences, one for encryption process and the second for embedding process.
2. A new LSB insertion technique suggested called random control bit technique.
3. In this paper, the basic efficient criteria are applied to test the efficiency of the proposed crypto-hiding key generator, and applying the steganalysis tested using visual, Laplace and chi-square tests, to test the RCB system.

2. Hiding in Images

In essence, image steganography is about exploiting the limited powers of the Human Visual System (HVS) [6]. Within reason, any plaintext, ciphertext, other images, or anything that can be embedded in a bit stream can be hidden in an image. To a computer, an image is an array of numbers that represent light intensities at various points, or pixels. An image size of 640 by 480 pixels, utilizing 256 colors (8) bits per pixel) is fairly common. Such an image would contain around 300 kilobits of data. Digital images are typically stored in either 24-bit or 8-bit per pixel files. 24-bit images are sometimes known as true color images. Obviously, a 24-bit image provides more space for hiding information; however, 24-bit images are generally large and not that common [7].

3. Least Significant Bit (LSB) Insertion Technique [8]

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. Any changes in the pixel bits will be indiscernible to the human eye. When using LSB techniques on 8-bit images, more care needs to be taken, as 8-bit formats are not as for giving to data changes as 24-bit formats are. Care needs to be taken in the selection of the cover image, so that changes to the data will not be visible in the stego-image.

4. Steganalysis

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 [9].

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.

Some steganographic utilities use secret keys. We can distinguish two kinds of keys: steganographic keys and cryptographic keys. A steganographic key controls the embedding and extracting process.A cryptographic key, however, is used to encrypt the message before it is embedded [10].

5. Classification of Steganography Attacks [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 (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. 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.

An **active attacker**, who is not able to extract or prove the existence of a secret message, thus can simply add random noise to the transmitted cover and so tries to destroy the information. In the case of digital images, an attacker could also apply image processing techniques or convert the image to another file format. Another practical requirement for a steganography system, therefore, is robustness. A system is called **Robust** if the embedding information cannot be altered without making drastic changes to the stego-object.

6. Steganalytic Methods

6.1 Visual Analysis [9]

The visual attack is a stego-only attack that exploits the assumption of most authors of steganography programs that the LSBs 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.

6.2 Statistical Analysis

When a steganography program embeds a bit through overwriting the LSB of a pixel is changed to an adjacent color value in the palette (or in the RGB cube if the cover file is a true-color image). These simple tests are not able to decide automatically if an image contains a hidden message. Now we will discuss the stego-only attack using chi-square test [12]: 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 LSB's 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 [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 χ2-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 [12]:

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

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

$$y_i = n_{2i} \qquad \ldots(3)$$

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

$$\chi^2 = \sum_{i=1}^{v+1} \frac{(y_i - y_i^*)^2}{y_i^*} \qquad \ldots(4)$$

where v 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. For an image that does not contain any hidden information, we expect the probability of embedding to be zero everywhere [12].

7. Stream Cipher systems [13]

In **stream ciphers**, the message units are bits, and the key is usual produced by a **random bit generator**. The plaintext is encrypted on a bit-by-bit basis. The key is fed into random bit generator to create a long sequence of binary signals. This “key-stream” k is then mixed with plaintext m, usually by a bit wise XOR (Exclusive-OR modulo 2 addition) to produce the ciphertext stream, using the same random bit generator and seed.

A **linear feedback shift register** is made up of two parts: a shift register and a **feedback function**. The shift register is a sequence of bits, (the length of a shift register is figured in bits). Each time a bit is needed, all of the bits in the shift register are shifted 1 bit to the right. The new left-most bit is computed as a function of the other bits in the register. The output of the shift register is 1 bit, often the least significant bit.

8. Basic Efficiency Criteria (BEC) of Key Generators

As known before, any stream cipher key generator consists of two basic units; they are sequence(s) of bit stream and **C**ombining **F**unction (CF) for the Key Generator (KG). The **basic criteria** of key generator efficiency can be defined as the ability of key generator and its sequence to withstand the mathematical analysis which the cryptanalyst can be applied on them [14].

Let key generator consists of n input sequences S_i, $1 \le i \le n$, let $S=\{s_0,s_1,\ldots\}$ be the output sequence which product from the key generator and s_j, j=0,1,… represents the elements j of S.

8.1 Randomness Criteria

Golomb′s randomness postulates [15] are presented here for historical reasons they were one of the first attempts to establish some necessary conditions for a periodic pseudorandom sequence to look random. Let $S=s_0,s_1,s_2\ldots$. be an infinite sequence. The subsequence consisting of the first n terms of S is denoted by $S_n=s_0,s_1,s_2,\ldots,s_{n-1}$. The sequence S is said to be **n-periodic** if $s_i=s_{i+n}$ for all $i \ge 0$.

Let S be a sequence. A **run** of S is a subsequence of S consisting of consecutive 0′s or consecutive 1′s which is neither preceded nor succeeded by the same symbol. A run of 0′s is called a **gap**, while a run of 1′s is called a **block**.

The **autocorrelation function** C(t) measures the amount of similarity between the sequence S and a shift of S by t positions. If S is a random periodic sequence of period n, then n.C(t) can be expected to be quite small for all values of t, 0<t<n.

Let S be a periodic sequence of period n. Golomb′s randomness postulates are the following [15]:

R1: In the cycle S_n of S, the number of 1′s differs from the number of 0′s by at most 1.

R2: In the cycle S_n at least half the runs have length 1, at least one-fourth have length 2, at least one-eighth have length 3, etc., as long as the number of runs so indicated exceeds 1. Moreover, for each of these lengths, there are (almost) equally many gaps and blocks.

R3: The autocorrelation function C(t) is two-valued. That is for some integer K:

$$n.C(t)=\sum_{i=0}^{n-1}(2s_i-1)\cdot(2s_{i+t}-1)=\begin{cases} n & ,t=0 \\ K & ,1\le t\le n-1 \end{cases} \qquad \ldots(5)$$

A binary sequence which satisfies Golomb′s randomness postulates is called a **pseudo-noise sequence** or a **pn-sequence**. Pseudo-noise sequences arise in practice as output sequences of maximum-length linear feedback shift registers.

8.2 Periodicity Criteria

Let Per(S) represent the period of the sequence S, let Per(S_i) be the period of the sequence S_i, $1 \le i \le n$, then [16]:

$$\text{Per}(S)=\text{lcm}(\text{Per}(S_1),\text{Per}(S_2),\ldots,\text{Per}(S_n)) \qquad \ldots(6)$$

Of course if Per(S_i) are relatively prime to each other $\forall i$, $1 \le i \le n$, then

$$\text{Per}(S)=\prod_{i=1}^{n}\text{Per}(S_i) \qquad \ldots(7)$$

8.3 Linear Complexity Criteria

One important metric used to analyze LFSR-based generators is Linear Complexity (LC). This is defined as the length, n, of the shortest LFSR (which is equivalent LFSR) that can mimic the generator output [16]. LC is important because a simple algorithm, called the Berlekamp-Massey algorithm, can generate this LFSR after examining only 2^n bits of the key stream [17]. Let F_n=CF, so that in general LC(S)$\le F_n^*(r_1,r_2,..,r_n)$, where F_n^* is the integer function corresponding to F_n, s.t. $F_n^*:Z^+ \to Z^+$, and r_i is the length of the equivalent LFSR which generate the sequence S.

8.4 Correlation Immunity Criteria

Siegenthaler has shown that Correlation Immunity (CI) can be precisely defined, and that there is a trade-off between CI and linear complexity [18].

Correlation can be defined as the relation between the sequence of CF=F_n from the key generator and the sequences that are combined each other by CF. This relation caused because of the non-linearity of the function F_n. The **Correlation Probability** (CP) of x, in general, represents the ratio between the number of similar binaries (N_s) of two sequences to the length (L) of the compared part of them [19].

$$CP=\frac{N_s}{L} \qquad \ldots(8)$$

F_n has mth order CI, if the output z of F_n is statistically independent from m output from m-sequences ($x_1,x_2,...,x_m$), of n combined sequences s.t. $m \le n$, in another word, if the CP approximate equal to 0.5 for m-sequences. The CI order can be calculated from the logical truth table of CF depending on calculating CP(x). The best CI for any system when m=n, that's mean all x_i, $1 \le i \le n$, are independent form the output z [18].

9. Statistical Randomness Tests

Designer and users of encryption algorithms used in cipher systems need a systematic approach in examining their cipher prior to use, to ensure that they are safe from cryptanalytic attack. In this manner we will introduce a new package of randomness instead of the mentioned five tests. CRYPT-X [20] is a microcomputer package that is intended to be used to test either large binary strings that are to be used as keystream in stream ciphers or block cipher algorithms. The package uses a number of statistical distributions including the **standard normal** and the **chi-squared distribution** [21].

The security of a stream cipher depends on the keystream appearing random. Tests employed in the package to examine this hypothesis are the **Frequency** tests on the original stream and the 1st and 2nd **Binary Derivative** stream, **Change Point** test, **Subblock** (**Poker**) test and **Run** test. The details of these tests are as follows:

10. Design and Implementation of Robust Stego-System

This paper aims to apply a new information hiding technique using LSB in digital BMP files hybrid with a new suggested technique. We will introduce the embedding and extracting algorithms supported by encryption algorithm based on a key generator for the encryption and embedding purposes, with experimental examples.

10.1 Design of Crypto-Hiding Key Generator Algorithm

The idea of construct single key generator with two outputs or purposes will be introduced; first it can be used as crypto key generator to encrypt (decrypt) the message want to be hiding (extracting). The second, and in the same time, a hiding key generator to specify the byte from the stego-image to be hide in.

The proposed key generator called Crypto-Hiding Key Generator (CHKG) which it's the heart of the steganography algorithm will be introduced.

10.1 Key Management of CHKG

Two kinds of keys considered as initial key or LFSR's initial values are suggested to use, the third is a value related to CHKG implementation. These keys are:

1. **Basic Key (BK)**: this key is secret which is consists of (20) ASCII code characters. This key distributed in classified document or protected computer file to keep it away from intruders. New BK used with every new message to guarantee no two messages has the same key. The used BK cancelled automatically in order not be used again.
2. **Message Key (MK)**: this key could be public which is consists of ASCII code (10) characters. This key generated automatically by the proposed system and sent in the same stego-image. MK will mixed with BK to increase the randomness of BK. If MK sent secretly, it will increase the key space of the CHKG.
3. **Maximum Jump (MJ)**: this key is an integer value ranged (12..32) represents the maximum difference between the current embedded bit and the next. This key as secret as BK and saved in the same document of BK.

10.2 Basic Components of CHKG

The proposed CHKG consists of the following main components:

1. **Main LFSR System (MLFSRS):** consists of (4) LFSR's [17] of lengths, 37,39, 41 and 47 respectively.
2. **Balance LFSR (BLFSR)**: is of length 53 bits.

3. **Combining Function (CF)**: we suggest to use the following non-linear function:

   $F(x_1,x_2,x_3,x_4) = x_1x_2 \oplus x_1x_3x_4 \oplus x_4$, where x_i is the output of $LFSR_i$, i=1,2,3,4.

4. **Control Octal (CO)**: Its represents the most 3 bits from the output of key generator byte to determine the decision bit.
5. **Binary to Integer (B2I) Converter**.
6. **MLFSRS Fixed Positions (p_i)**: (4) fixed stage positions are needed, each from one LFSR, the proposed position is the stage number (23) from each one such that: $p_i = LFSR_i(23)$, i=1,2,3,4.
7. **BLFSR Fixed Positions (bp)**: (8) fixed stage positions are needed, the proposed positions are the following stages: bp = BLFSR(13,17,23,29,31,37,43,47).

10.3 CHKG Initialization

The main steps of CHKG initialization are:

1. Every character of BK transform to (8) bits, then the string of BK (STBK) has length (20*8=160 bits): $STBK=BK_1\ldots BK_{20}=BK_{1,7}\ldots BK_{1,0}BK_{2,7}\ldots BK_{2,0}\ldots BK_{20,7}\ldots BK_{20,0}$.

   where BK_i is the BK character number i, 1≤i≤20, and BK_{ij} is the bit j of BK_i, 0≤i≤7.

2. In the same way we get STMK which has length (10*8=80 bits):

   $STMK=MK_1\ldots MK_{10}=MK_{1,7}\ldots MK_{1,0}MK_{2,7}\ldots MK_{1,0}\ldots MK_{10,7}\ldots MK1_{10,0}$.

   where MK_i is the MK character number i, 1≤i≤10, and MK_{ij} is the bit j of MK_i, 0≤i≤7.

3. The string of the initial key (STIK) of MLFSRS is the XOR sum of STMK and STBK:

STMK	=	M_1	M_2	M_3	…	M_{80}	M_1	M_2	M_3	…	M_{80}
STBK	=	B_1	B_2	B_3	…	B_{80}	B_{81}	B_{82}	B_{83}	…	B_{160}
STIK	=	I_1	I_2	I_3	…	I_{80}	I_{81}	I_{82}	I_{83}	…	I_{160}

   Where $I_j=M_j$ xor B_j, M_j, B_j and I_j are the bits j of STMK, STBK and STIK respectively.

4. From STIK the LFSR's of MLFSRS are filled from STIK one by one and stage by stage as follows: $LFSR_1$: STIK (1-36), $LFSR_2$: STIK (37-74), $LFSR_3$: STIK (75-114), $LFSR_4$: STIK (115-160). The last stage of each LFSR filled with (1).
5. The BLFSR can be filled when the MLFSRS start move, such that this system move 52 movements using the CF to fill BLFSR s.t.:
   $BLFSR(i)= F(x_{1i}, x_{2i}, x_{3i}, x_{4i})$, i=1,2,…,52. The last stage of BLFSR filled with (1).

10.4 CHKG Movement

The CHKG movement can be summarized by the following steps:

1. The MLFSRS start move, we get the outputs x_i, as address to the CF to get Z, s.t. $Z=F(x_1,x_2,x_3,x_4)$.
2. Let the output of BLFSR be x_5, then the Crypto-Key bit (CKb) is: $CKb = Z \oplus x_5$.
3. Let X be the output of B2I converter function using the output of MLFSRS s.t.
   $X = B2I\ (x_1,p_1,\ x_2,\ p_2,\ x_3,\ p_3,\ x_4,\ p_4)$.
4. Let Y be the output of B2I converter function using the output fixed positions of BLFSR s.t. $Y = B2I\ (bp_1,\ bp_2,\ bp_3,\ bp_4,\ bp_5,\ bp_6,\ bp_7,\ bp_8)$.
5. Let the Embedding Key Byte (EKB) defined as follows: $EKB = X \oplus Y$
   This key divided into two parts:
   (a). The most (3) bits are represent the Control Octal (CO) s.t.
   $CO = EKB_7*4 + EKB_6*2+ EKB_5$, (if CO=0, then CO=1)
   (b). The Embedded Key Jump (EKJ) can obtained as follows: EKJ = EKB mod MJ,

Since max{MJ} is 32, this mean the last (5) bits of EKB are used.
The block diagram of CHKG is shown in figure (1).

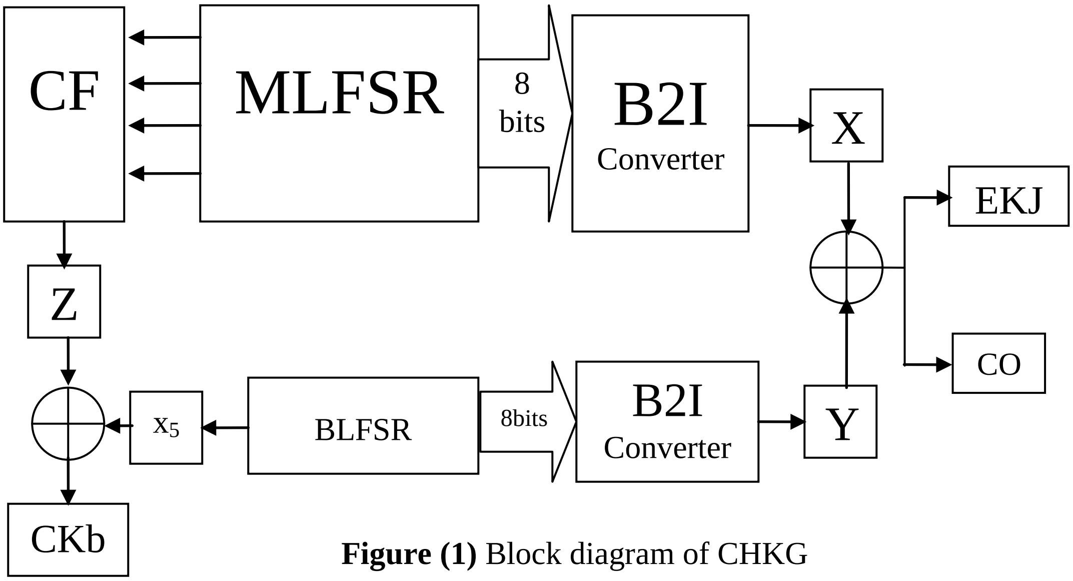


Figure (1) Block diagram of CHKG

11. Implementation of Basic Efficient Criteria on CHKG

11.1 Randomness Criterion (R)

Notice that F is balance since number of output 0’s equal to 1’s, so the randomness statistics results of CHKG will expect to be real random. We have to test the randomness of the CHKG in two ways. First, we have to test the randomness of the crypto part (using sample c with length 21008 bits). Second, we have to test the hiding part (using sample h with length 21008B (=168064 bits)).

CRYPT-X [51] package used to test the output results of CHKG tested by using group of tests. Table (1) shows the Frequency test, 1st and 2nd Binary Derivative test.

While Change Point test shown in table (2). Subblock (Poker) and Run tests described in table (3).

Table (1) CHKG tested by frequency and binary derivative tests.

Test	Ex.	Length	n_1	mean(1)	prop(1)	α
Freq.	c	21008	10940	10504.0	0.4993	0.8468
	h	168064	84218	84032.0	0.5011	0.3642
1st BD	c	21008	10506	10503.5	0.5001	0.9725
	h	168064	83908	84031.5	0.4993	0.5468
2nd BD	c	21008	10579	10503.0	0.5036	0.2943
	h	168064	84300	84031.0	0.5016	0.1894

Table (2) CHKG tested by Change point test.

Ex.	Length	Change point (Cp)	n_1 before Cp	prop(1) before Cp	prop(1) after Cp	α
c	21008	8858	4377	0.4941	0.5031	0.4452
h	168064	71712	35836	0.4997	0.5021	0.6250

Table (3) CHKG tested by subblock and Run tests.

Test	Ex.	Size	χ^2-value	DF	mean(1)	prop(1)	α
Subblock	c	2	1.2338	3	-------	-------	0.7449
	h		2.2141				0.8580
	c	4	9.5872	15	-------	-------	0.8449
	h		10.7923				0.7923
	c	8	246.6523	255	-------	-------	0.6389
	h		241.5842				0.7190
Run	c	-----	13.4825	22	5253	5254	0.9188
	h		17.2414	28	41955	41954	0.9437

Where $\mathbf{n_1}$ denotes the number of ones, **mean(1)** is the mean of ones, **prop(1)** is the proportion of ones, and lastly, **α** is the tail-area probability.

11.2 Periodicity Criterion (Per)

$$\text{Per(CHKG)} = \text{lcm}\ (\text{LFSR}_1, \text{LFSR}_2, \text{LFSR}_3, \text{LFSR}_4, \text{BLFSR}) \qquad \ldots(9)$$

$$=\text{lcm}\ (2^{37}-1, 2^{39}-1, 2^{41}-1, 2^{47}-1, 2^{53}-1) \approx 2^{53} = 9007199254740992.$$

11.3 Linear Complexity Criterion (LC)

This criterion depends on the length of combined LFSR's and the CF of the generator, the suggested CF is 3rd order non-linear function.

$$\text{LC(CHKG)} = \text{LC(MLFSRS)} + \text{LC(BLFSR)} \qquad \ldots(10)$$

$$= F^*(37, 39, 41, 43)+53=37*39+37*41*47+47+53 = 72841.$$

Where F^* is the integer value of the logical function F.

Now BerleKamp-Massey algorithm will applied to estimate the LC of CHKG, Table (4) shows LC(CHKG) for various sequences length from the two outputs.

Table (4) LC(CHKG) of various sequences length.

Ex.	**Length**	**output**	**LC(S)**
2	10000	Crypto	5003
		Embedding	619
3	20000	Crypto	10007
		Embedding	1249

11.4 Correlation Immunity Criterion (CI)

This criterion depends on CF only. The CP for the values x_1, x_2, x_3, and x_4 with output of CF is 0.5, 0.625, 0.5 and 0.75 respectively, therefore CI (CHKG) = 2. So its weak! That what make us to add the 5th LFSR, which be called balance LFSR, we will notice that the CP will equal 0.5 for all inputs, then: CI (CHKG) =5= Number of LFSR's.

Table (5) shows these results from various comparison sequences lengths (the shaded cells have CP≠0.5).

Table (5) CP and CI results for various sequences lengths.

Length in bits	**CP**					**Function**	**CI**
	x_1	**x_2**	**x_3**	**x_4**	**x_5**		
11920	0.502	0.630	0.503	0.745	-----	Z	2
	0.494	0.497	0.500	0.498	0.499	CKb	5
21008	0.503	0.628	0.502	0.746	-----	Z	2
	0.499	0.499	0.502	0.498	0.499	CKb	5

12. Design of Random Control Bit Stego-System

In this manner the design of the proposed steganography system will be shown. First, a new technique introduced by developing the LSB insertion technique, second, the embedding and extracting will be described for one bit from the ciphertext. Lastly, we will show the extension process of the data embedding in whole the image.

12.1 Random Control Bit Technique

The proposed steganography system using the LSB technique, so when the hiding byte is specified, which is consists of (8) bits (7..0), so the LSB bit used to be hide in. Let's denote the Embedded bit by (Eb), so LSB=Eb. Here, a new technique be suggested which be called the Random Control Bit (RCB) technique. This technique summarized by choosing another bit from the Byte of Embedding (BE) different from the LSB randomly and its ranged (1..7). The new chosen bit called the Decision bit (Db). The idea of embedding is said that "**if Db=Eb then the LSB=0, else LSB=1**". The Db bit called **decision bit** since it has the decision to make LSB 0 or 1 when it compared with Eb bit. While the extracting idea is "**if LSB=0 then Eb=Db, else Eb=complement of Db**".

Let us formulates the ideas in pseudo code logical equations, so the hiding equation is:

```
IF Eb = Db THEN LSB := 0  ELSE LSB := 1;                    …(11)
```

The extracting pseudo code logical equation is:

```
IF LSB = 0 THEN Eb := Db  ELSE Eb:= Db XOR 1;               …(12)
```

The new idea aims to combining another bit from the byte of embedding with LSB to increase the complexity of hiding.

Equations (4-3) and (4-4) can be reformulated without using "**IF**" statement by equations (4-5) and (4-6) respectively:

LSB = Eb XOR Db …(13)

And

Eb = LSB xor Db …(14)

The Db can be specified from the following equation:

Db = EB [CO] …(15)

Figure (2) shows the details of RCB process.

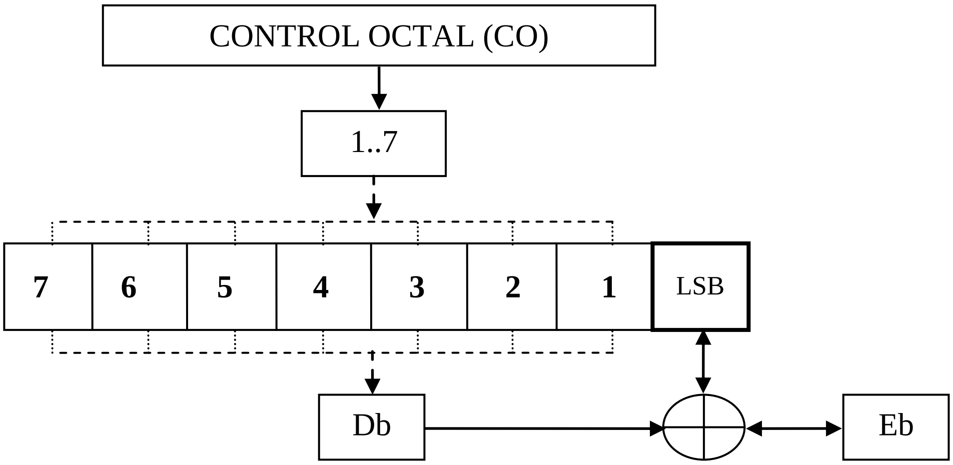


Figure (2) Random Bit Control (RCB) technique

12.2 Embedding and Extracting of Cipher Bit

The proposed system depends on CHKG which act in two functions. The first function represents the crypto process. If P is the plaintext of the message with length n which want be hide, then its break into binary data m_1, m_2,…, m_n. If the ciphertext of the message is c_1, c_2,…, c_n, then the encryption process is:

c_i = CKb_i XOR m_i, i=1,2,…,n …(16)

now the bit want be embed is c_i, then

$Eb_i = c_i$ …(17)

If we apply equation (13) to embed c_i, so the embedded process is:

$LSB_i = Eb_i \text{ XOR } Db_i$ …(18)

The extracting equation derived from equation (14) such that:

$Eb_i = LSB_i \text{ XOR } Db_i$ …(19)

Since $c_i = Eb_i$, then:

$m_i = c_i \text{ XOR } CKb_i$ …(20)

12.3 Extending the Embedding Process

Let's denote the pure size of stego image (the pure size is the size of the image without the header information) by S(I) in bytes, the size of the hidden message by S(M) in bits. So we can defined the Extended Jump (EJ) as the maximum jump or an upper bound of the jump that can guarantees the message can be hiding in the image normally.

A relation between the extended jump and the size of the message can found, assuming that the size of the cover image is constant, s.t.

$$EJ \propto \frac{1}{S(M)}, \text{ Then } EJ = S(I) \text{ DIV } S(M) \qquad \text{…(21)}$$

The proposed steganography algorithm will be designed to use a random generator in order to produce random jumps in the stego-image, so the CHKG has its own jump we call it embedding key jump (EKJ). The EKJ is ranged $0 \leq EKJ \leq MJ$, where MJ is the maximum jump could be done by CHKG.

Always $EJ \geq MJ$, if it converse, that means the cover image can't occupy the message. In general to guarantee hiding the message in the cover image using LSB technique, must be: $S(I) \geq S(M)^* MJ$.

Let the Expected Used Size (EUS) when using MJ is: $EUS = S(M)^*MJ$

The Expected Proportion (EP) of the expected used size to the size of image:

$EP = EUS/S(I) = MJ^*(S(M)/S(I)) = MJ/EJ$

We suggest using threshold of accepting the embedding (the suggested threshold is 0.75, means the embedded message distributes in 75% of image size), then if $EP \leq$Threshold then we continue the jump extension process, else we need no extension.

Now we defined the Desired Jump (DJ) as the jump which needs to add it to the MJ in order to get EJ, so:

$DJ = EJ - MJ$ …(22)

DJ = 0 when need no extension. Then:

$RJ = DJ + EKJ$ …(23)

Where RJ is the Real Jump or new key jump, s.t. $DJ \leq RJ \leq EJ$.

This work here is to extend the EKJ to be RJ, s.t. $RJ \geq EKJ$ in order to distribute the message bits in the entire cover image and not to be specified in part or subarea from the image which it is clear to the VHS.

Example (1):

Let us have the following information:

S(M)=0.625KB= 640B= 5120 bits, S(I)= 500KB= 512000B, MJ = 32B.

$\therefore$ EJ = S(I) div S(M) = 100B.

EUS = S(M)*MJ = 5120*32 = 16384B.

EP = MJ/EJ =32/100 = 0.32.

Let the suggested threshold is 0.75, then if EP ≤ 0.75 then

DJ = EJ – MJ = 100 – 32 = 68.

RJ = EKJ + DJ = EKJ + 68.

Since 0≤EKJ≤MJ=32 then 68=DJ≤RJ≤EJ=100.

13. Detection Tests of RCBS System

13.1 Applying the Detection Tests

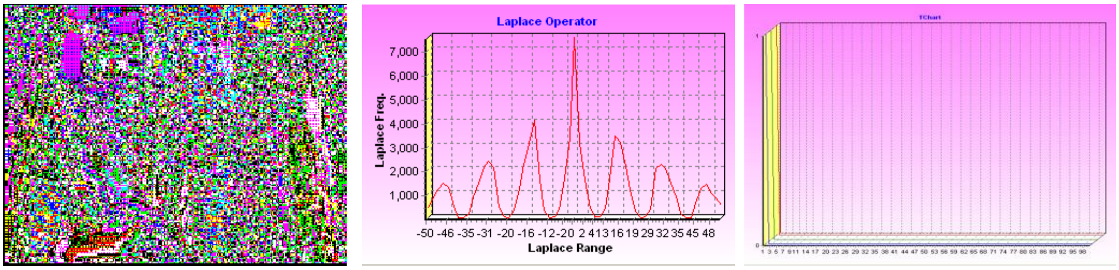
In this section we apply three kinds of detection steganalytic tools, these tools are visual attack, Laplace Operator and Chi-Square test. First, the image before embedding will be shown, then we will embed a message with sequential embedding and last the same message will embedded with RCBS system. The three tests are applied on the three images to show the difference between the sequential and suggested hiding systems.

Figure (3) shows the three images (before, sequential and RCBS embedding systems) as VHS.

(a). Before embedding (b). Sequential (c). RCBS system

Figure (3) Delphi.bmp images before, sequential and RCBS embedding.

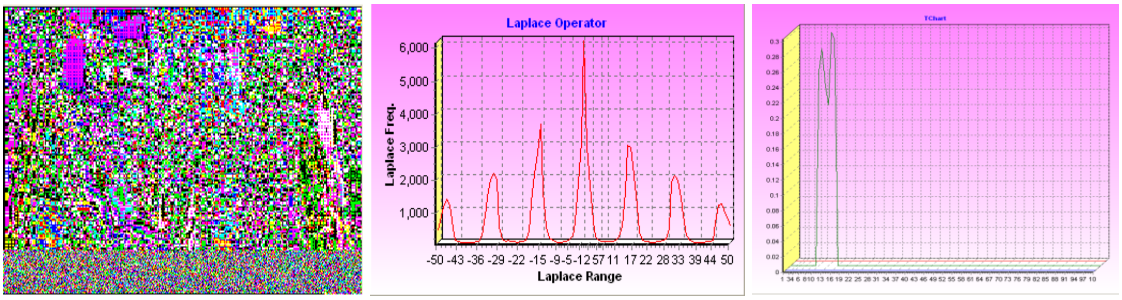
In figure (4) the results of the (3) tests done in Delphi.bmp which has no embedding text.



(a). Visual (b). Laplace operator (c). Chi-square test.

Figure(4) Delphi.bmp with visual, Laplace and Chi-square tests without hiding.

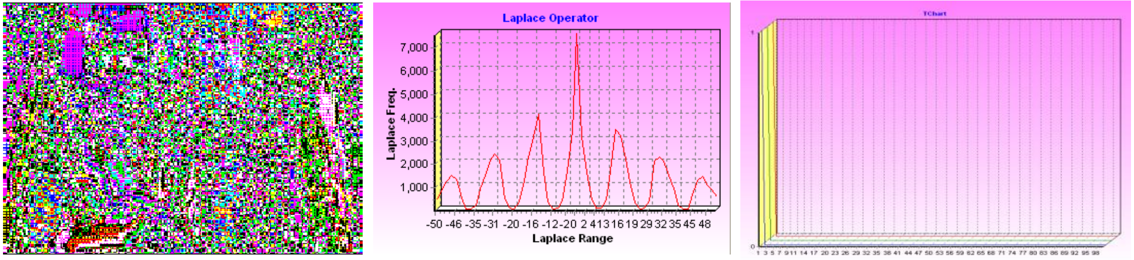
In figure (5) the results of the (3) tests done in Delphi.bmp which has embedding information using sequential embedding.



(a). Visual (b). Laplace operator (c). Chi-square test.

Figure(5) Delphi.bmp with visual, Laplace and Chi-square tests by sequential embedding.

In figure (6) the results of mentioned three tests done in Delphi.bmp which has embedding information using RCBS.



(a). Visual (b). Laplace operator (c). Chi-square test.

Figure(6) Delphi.bmp with visual, Laplace and Chi-square tests using RCBS.

13.2 Detection Tests Results Analysis

In this section we will detail the analysis interpretation of the detection tests applied in section (13.1). One image (Delph.bmp) chosen to be the practical example, the comparison study consists of two kinds of embedding system; the sequential and the RCBS.

The pictures in figure (4) represent the three detection tests results of the image which has no embedded information. These pictures can be considered as a measure to other pictures tests in the rest figures. If the pictures of the tests in other figures are not exactly similar to the pictures shown in figure (4), then the image may contains hidden information.

Figure (5) deals with sequential embedding. Image (a) of figure (5) represent the visual test, more condense points are noticed in the bottom of the picture which is different from image (4-a). Compare image (5-b) of Laplace test, with image (4-b) a little difference noticed in the XY-dimensions. image (c) of figure (5) of chi-square test signs high peaks in the left side of the picture which has clear difference than image (4-c). Figure (6) deals with RCBS system .All the images of figure (6) are exactly the same as pictures of figure (4). Delphi.bmp seems has no embedded information.

14. Conclusions and Future Works

1. The suggested threshold relies by the available steganalytic studies and the detecting tools which are can be done in image analysis.
2. In section (13-2) we discuss the tests analysis on stego-image compared with cover-image which has no hidden data yet. But, we have to discuss the tests analysis of stego-image without comparison. the stego-image has condense points in visual attack, peaks appearance in the curve of Laplace attack, while the curve of chi-square has no smooth line as in curve of the original image.
3. It’s important to mention that the detection tests not always gives accuracy decision about the existence of the hidden data in the attacked image.

This thesis recommends the following points enhance the work of the RCBS system:

4. The RCB technique depend on LSB bit only, we think it can be using two least bits to increase the hidden message length.
5. The extension process used in RCBS system is linear extension, it can be developed to be non-linear by using the following two equations:

$$DJ = EJ \text{ DIV } MJ \qquad \ldots(24)$$

$$RJ = DJ*EKJ \qquad \ldots(25)$$

These equations will increase the range of RJ to be $0 \leq RJ \leq EJ$ instead of $DJ \leq RJ \leq EJ$.

6. We suggest add the detection tests to the RCBS as an option act after embedding part to give a decision about if the stego-image passes the detection tests successfully.

References

[1]. Bourke, P. “***BMP Image Format***”, Englewood cliffs: Prentice-hall, July 1998.4

[2]. Wu, N., and Hwang, M., “Data Hiding: Current Status and Key Issues”. International Journal of Network Security. (2007, Jan.) Available:www.ijns.nchu.edu.tw/ijnsv4-n1/ijns-2007-v4-n1-p1-9.pdf.7

[3]. Pal, S. K., Saxena, P. K. and Mutto, S. K., “A Systematic Approach to Stegangraphic of Images”, URL, 2002.8

[4]. Al-hamami. Mohammed, “Information Hiding Attack in Image”, M. Sc. thesis introduced to Ministry of Higher Education & Scientific Research Iraqi Commission for Computer & Informatics, Informatics Institute for Postgraduate Studies, 2002.9

[5]. Alwan, R., Kadhim, F., and Al-Taani, A., “Data Embedding Based on Better Use of Bits in Image Pixels”. International Journal of Signal Processing. [Online]. (2005) Available: http://www.enformatika.org/ijsp/v2/v2-2-15. pdf. 10

[6]. Salima B. Abdulah and Hilal M. Yousif, “Arabic Text Information Hiding”, M.Sc. Thesis, Iraqi Commission for Computers and Informatics Institute for Postgraduate Studies, 2003.3

[7]. Johnson,N.F.,“Steganography”,WWW:http://www.jjtc.com/stegdoc/.George Mason University, 2003.6

[8]. Chandramouli, R. and Memon, N., “Analysis of LSB Based Image Steganography Techniques”, presented at IEEE International Conference on Image Processing (ICIP), Oct 7-10 2001, Thessaloniki, 2001.11

[9]. Sellars,D.,“AnIntroductionto Steganography”, www.cs.uct.ac.za/.../stego.ps.gz,99.5

[10]. Memon Nasir D., Khalid Sayood, “Lossless Compression of Color Image in the RGB Domain”, Computer Science and Mathematics, Arkansas State University, USA, 2001.28

[11]. Fridrich J. and Miroslav G. and Hogea D., “Steganalysis of JPEG Images: Breaking the F5 Algorithm”, SUNY Binghamton, Dept. of Computer Science, Binghamton, NY 13902-6000, USA, 2001.29

[12]. Rifat Z. K. “Statistical Approach for Steganalysis”, M.Sc. Thesis, Applied Sciences of University of Technology 2003.30

[13]. Johnson, N. F., Duric, Z. and Jajodia, S., “Information Hiding: Steganography and Watermarking-Attacks and Countermeasures”. Kluwer Academic Publishers, Boston Dodrecht London, 2000.18

[14]. Yan, S. Y., “Number Theory for Computing”, Springer-Verlag Berlin Heidelberg, New York, 2000.21

[15]. Stinson, D. R., “Cryptography: Theory and Practice” CRC Press, 1995.35

[16]. Golomb, S. W., “Shift Register Sequences” San Francisco: Holden Day 1982.

[17]. Apostol, T. M., “Introduction to Analytic Number Theory”, Corrected 5th Printing, Undergraduate Texts in Mathematics, Springer-Verlag, 1998.48

[18]. Schneier, B., “Applied Cryptography (Protocols, Algorithms, and Source code in C.)” Second Edition 1997”, John Wiley & Sons Inc.37

[19]. Baum, U. and Blackburn, S. “Clock Controlled Pseudorandom Generators on Finite Groups”, K.U Leuven Workshop Cryptographic Algorithms, Springer-Verlag, 1995.51

[20]. Rueppel, R. A., “Analysis and Design of Stream Ciphers” Springer-Verlag, 1986.47

[21]. Gustafson, H., Dawson, E. “A Computer Package for Measuring the Strength of Encryption Algorithm”, Information Security Research Centre at Queensland University of Technology, 1998.52

[22]. Bluman, A. G., “Elementary Statistic: Step by Step Approach”, 6th ed., McGraw-Hill Companies Inc., New York, NY10020, 2007. 53