Semi Random Position Based Steganography for Resisting Statistical Steganalysis

Amitava Nag[1], Sushanta Biswas[2], Debasree Sarkar[2], and ParthaPratim Sarkar[2]
(Corresponding author: Amitava Nag)

Department of Information Technology, Academy of Technology, Hoogly 721212, India[1]
Département of Engineering and Technological studies,University of Kalyani[2]
Kalyani 741 235, India
(Email: amitavanag.09@gmail.com)
(Received Oct. 18, 2012, revised and accepted Feb. 20, 2013)

Abstract

Steganography is the branch of information hiding for secret communication. The simplest and widely used steganography is the LSB based approach due to its visual quality with high embedding capacity. However, LSB based steganography techniques are not secure against statistical steganalysis mainly $\chi 2$ attack and Regular Singular (RS) attack. These two staganalysis can easily estimate the hidden message length. This work propose a LSB based steganography technique where first a location is obtained randomly based on the bit pattern (except LSB) of a cover pixel using linear probing and embed a secret bit into LSB. This technique makes the stego-image completely secure against both $\chi 2$ attack and RS attack.

Keywords: Information hiding, RS attack, statistical steganalysis, steganography, $\chi 2$ attack

1 Introduction

Due to widespread use of internet, the sharing and transmission of images in digital form has become quite easy. However, message transmissions over the Internet still have to face all kinds of security problems. Therefore, finding ways to transmit data secretly through internet has become an important issue. Cryptography [18] is a one procedure to provide a safe way by transforming data into a cipher text via cipher algorithms [1, 15]. Encryption techniques scramble the message so that it cannot be understood by unauthorized users. However, this can naturally raise the curiosity level of an eavesdropper. It would be rather more prudence if the secret message is cleverly embedded in another media such that the secret message is concealed to everyone. This idea forms the basis for steganography [3, 13], which is a branch of information hiding by camouflaging secret information within covert carriers to avoid observation. The word steganography in Greek means "covered writing". Steganography is the art of hiding the presence of communication by embedding secret messages into innocent, innocuous looking cover documents, such as digital images [2, 4, 7, 8, 10, 12, 20, 21], videos [5, 19], sound [10, 14, 16] or document [6, 11, 17]. The stego-medium is the result of embedding the message in cover-medium. Images provide excellent carriers for hidden information. Many different techniques have been introduced to embed messages in images.

The most common approaches for steganography in images are Least Significant Bit (LSB) modification [8, 12] where LSB is substituted by secret bit. This modification create some structural asymmetry [8, 13] and thus sufficient evidence about the existence of secret message inside stego-image is collected. This is known as steganalysis attack [13, 20]. The goal of the proposed method is to avoid detection of steganalysis attack.

2 Related Work

Steganalysis is the science and art of discovering the existence of secret message hidden in stego-image using steganography [3, 13]. Steganalysis can be classified into two classes: signature steganalysis and statistical steganalysis [13, 20]. In statistical steganalysis the existence of the hidden message is discovered by finding statistical abnormality in stego-media caused by message embedding. Two popular statistical steganalysis are Chi-squared ($\chi 2$ detection) [21] and Regular Singular(RS) attack [4].

2.1 $\chi 2$ Detection

In the embedding process of LSB steganography the secret bits replaces the bits of LSB plane of cover image while the higher bit planes are unaltered. Hence, the pixel values of the cover Image are modified. This modification allow us to test Chi-squared (χ^2 detection) on the stego images. In westfeld and Pfitzmann proposed a chi-square detection (χ^2 detection) method using the POV (pair of value) phenomenon of the LSB plane.

The result of the LSB embedding process is the creation of Pairs of Value (POVs). For example, a pixel value of 210 in cover image will either 210 or change to 211. Thus

{210,211} is POV. In general, {2i,2i+1| $0 \le i \le 127$ } form POVs.

LSB embedding techniques creates POVs, the frequencies of 2k and 2k+1 becomes equal or nearly so. χ^2 attack detects these near-equal POVs and calculates the probability of embedding. The χ^2 statistics is calculated as

$$\chi^2 = \sum_{i=0}^{127} \frac{(x_i - z_i)^2}{z_i} where z_i = \frac{x_i + x_{i+1}}{2} \dots\dots\dots (1)$$

z_i is the theoretically expected frequency if a random message has been embedded, and x_i is the actual number of occurrences of color. The embedded rate p is estimated by the equation given below:

$$p = 1 - \frac{1}{2^{\frac{n-1}{2}}\Gamma\left(\frac{n-1}{2}\right)} \int_0^{x_{n-1}^2} e^{-\frac{u}{2}} u^{\frac{n-1}{2}-1} du \dots\dots (2)$$

But the main weakness of Chi-Squared technique is its complete dependency upon the pairs of values. This test fails completely for the any image embedding techniques using an algorithm that does not generate POVs.

2.2 RS Attack

Fridrichet. al. proposed the RS steganalysis technique for detecting embedded messages in a stego-image by LSB steganography. This method is most reliable and accurate method than χ^2 detection to estimate embedding rate for random embedding cases and can easily distinguish stego-image and cover image. The RS attack steganalysis technique perform the following steps

Step 1: Select an m-tuple Mask M with values {-1,0,1}. Here we choose m=4 and select M=[0 1 1 0], -M=[0 -1 -1 0].

Step 2: The grayscale image is divided into non-overlapping groups G_c of n adjacent pixels $x_1, x_2, \dots, x_n \in \{0 \text{ to } 255\}$ and set $G_c = (x_1, x_2, \dots, x_n)$

Step 3: The smoothness of pixel group G_c is determined by using the discrimination function f as

$$f(x_1, x_2, \dots\dots\dots, x_n) = \sum_{0}^{n-1} |x_{i+1} - x_i| \dots\dots\dots (3)$$

Step 4: An invertible mapping F_M, called flipping function is defined on [0 to 255] to flipping the pixel value according to M i.e. F_1 for positive M and F_{-1} for negative M as,

$$F_1(x) = \begin{cases} x-1, & if x \bmod 2 == 1 \\ x+1, & if x \bmod 2 == 0 \end{cases} \dots\dots\dots (4)$$

$$F_{-1}(x) = F_1(x+1) - 1 \dots\dots\dots (5)$$

where F_1: 0←→1, 2←→3, ..., 254 ←→255

and F_{-1}: -1 ←→0, 1 ←→2,3 ←→ 4, ..., 253 ←→ 254, 255←→256

Similarly, define F_0 as, $F_0(x) = x$

Step 5: Let $F_M(G_c)$ be the result of flipping all the pixel value of group G_c by flipping function F_M . Let $f(F_M(G_c))$ be the result of $F_M(G_c)$ input to discrimination function f, define three types of groups R,S,U by the following rules:

Regular Groups: $G_c \in R \Rightarrow f(F_M(G_c)) > f(G_c)$;

Singular Groups: $G_c \in S \Rightarrow f(F_M(G_c)) < f(G_c)$;

Unusable Groups: $G_c \in U \Rightarrow f(F_M(G_c)) = f(G_c)$.

Do this same grouping also for negative Mask i.e. –M.

Step 6:

a) Repeat Step 1 to Step 5 up to half of the total no of pixels of the embedded image. The factor of one half is due to the fact that, assuming the message is a random bit-stream; on average only one half of the pixels will be flipped.

b) Repeat Step 1 to Step 5 up to the total no of pixels of the embedded image.

Step 7: For half of the total no of pixels calculate, $R_M(p/2), S_M(p/2), R_{-M}(p/2), S_{-M}(p/2)$. Similarly for the total no of pixels calculate, $R_M(1-p/2), S_M(1-p/2), R_{-M}(1-p/2), S_{-M}(1-p/2)$, where p is unknown length of message in a stego-image (in percent of pixels).

Step 8: Obtain the value of x from the following quadratic equation,

$$2(d_1 + d_0)x^2 + (d_{-0} - d_{-1} - d_1 - 3d_0)x + d_0 - d_{-0} = 0 \quad \dots (6)$$

$$\text{where} \begin{cases} d_0 = R_M\left(\frac{p}{2}\right) - S_M\left(\frac{P}{2}\right) \\ d_1 = R_M\left(1 - \frac{p}{2}\right) - S_M\left(1 - \frac{P}{2}\right) \\ d_{-0} = R_{-M}\left(\frac{p}{2}\right) - S_{-M}\left(\frac{P}{2}\right) \\ d_{-1} = R_{-M}\left(1 - \frac{p}{2}\right) - S_{-M}\left(1 - \frac{P}{2}\right) \end{cases} \dots\dots\dots (7)$$

Step 9: p is estimated from x as $p = \frac{x}{\left(x - \frac{1}{2}\right)}$

2.3 RHTF-based LSB Steganography

In [11], the authors proposed a reversible histogram transformation functions (RHTF) steganography scheme. In 2011, Lou and Hu discovered two vulnerabilities: "zero points" and "double frequency", by which they distinguished the stego-images from cover images and also they broke the value of the secret key [9] . In [20], Lin and Hu remove "zero points" and "double frequency" from sego-image using the following algorithm:

Embedding Algorithm

Step 1: Divide the cover image I_c into n groups of size $\frac{I_c}{n}$.

Step 2: Generate n number of secret key a_j using following key generation algorithm:

a) Generate key by increasing a_j as $a_j = a_{j-1} + 1$, until $a_j = a_U$.

b) Generate key by decreasing a_j as $a_j = a_{j-1} + 1$, until $a_j = a_L$.

c) Continue Steps 2 (a) and (b) until j = n.
wherea initial value (a_1), upper bound(a_U) and lower bound (a_L) are predefined.

Step 3: Apply the compressing transformation technique to each cover pixel P as follows

$$P_1 = P - \left\lfloor \frac{P}{a_j + 1} \right\rfloor$$

Step 4: Replace the LSB of P_1 by a secret bit and P_2 is obtained.

Step 5: Apply the following formula to produce stego pixel

$$P_s = P_1 - \left\lfloor \frac{P_1}{a_j} \right\rfloor$$

Extracting Algorithm

Step 1: Divide the stego-image I_s into n groups of size $\frac{I_s}{n}$.

Step 2: Generate n number of secret key a_j using following key generation algorithm:

a) Generate key by increasing a_j as $a_j = a_{j-1} + 1$, until $a_j = a_U$.
b) Generate key by decreasing a_j as $a_j = a_{j-1} + 1$, until $a_j = a_L$.
c) Continue Step 2 (a) and (b) until j = n.
wherea initial value (a_1), upper bound(a_U) and lower bound (a_L) are predefined.

Step 3: Apply the compressing transformation technique to each cover pixel P as follows

$$P_1 = P_s - \left\lfloor \frac{P_s}{a_j + 1} \right\rfloor$$

Step 4: Extract the LSB of P_1 as a secret bit.

3 Proposed Technique

Sequential embedding is susceptible to Chi Square Attack [21]. In order to resist this, non-sequential random embedding was proposed. However, as a counter measure RS Attack [4] was developed specifically to detect random embedding. RS attack is based on the fact that image gets very slightly distorted as a result of LSB embedding. This approach groups a collection of pixels based on the effect embedding has on them as Regular, Singular and Unusable groups. Based on certain formulae and the number of Regular and Singular groups, we can deduce the percentage of embedding in terms of a probabilistic result. Here lies the strength of the RS algorithm.

The proposed algorithm takes advantage of the strongest point of the RS algorithm, i.e. it groups collection of pixels into Regular, Singular and Unusable groups. However, we find that only the Regular and Singular groups suffice the needs of the formulae. The proposed algorithm embeds the message bits in such a semi random way that the effects of embedding are cancelled out by each other and as a result the relative numbers of Regular and Singular Groups are minimized. Due to this, formulae render a misleading and impossible probabilistic result. Thus, the proposed algorithm resists both types of statistical steganalysis, namely Chi Square attack and RS Attack.

To insert a secret bit in the proposed technique, we first select a proper position. For position selection, we traverse the image according to the binary bit pattern of a pixel till its LSB is encountered. For traversing from location (x,y), we use the following rules:

Rule 1: If a 1 is encountered, then traverse the matrix by one row in a cyclic format, towards downward as $(x_{new}, y_{new}) = (((x mod M) + 1), y) \ldots\ldots (8)$.

Rule 2: If a 0 is encountered, then traverse the matrix by one column in a cyclic format, towards right side as $(x_{new}, y_{new}) = (x, (y\ mod N) + 1)) \ldots\ldots (9)$.

When a position of embedding $(x_{new}, y_{new})$is selected using above traversal, we check whether a collision occurs i.e. any bit is already embedded in that position. If collision occurs, then linear probing is used in row major order to select a collision free position for embedding a secret bit. After the selection of final position in cover image, the secret bit is directly embedded in the LSB of that selected cover pixel. To detect collision a binary flag matrix of same size of cover image is used whose elements are initialized to zero. When a secret bit is embedded in the cover image, the value in the corresponding position of the flag matrix is set to 1. During embedding in selected position of the cover image, if the value of corresponding position of flag matrix is 1, collision is detected and linear probing is applied until a position whose flag value is 0.

Algorithm 1:Embedding of secret bit stream

Input: Cover Image I of size M×N and Secret bits sequence S = {s1,s2,……,sL}, where si Є {0,1}
Output: Stego Image Is
1: Begin
2: Transformation of pixel value to even number: Change pixel I(x,y) as

$$I(x,y) = \begin{cases} I(x,y) - 1 & \text{if} I(x,y) \bmod 2 == 0 \\ I(x,y) & \text{Otherwise} \end{cases} \quad \ldots\ldots (10)$$

3: Binary pattern generation: Obtain binary value of I(x,y) as I(x,y)=(b7b6b5b4b3b2b1b0) where bi Є {0,1}.
4: Declaration and Initialization of Flag matrix: Declare a flag matrix F of size M×N and initialize its value with 0.
5: Image matrix traversal: Traverse the image matrix I according to the binary bit pattern of pixel I(x,y) starting from location (x,y) till the 2nd least significant bit (Here b1) using the following rules:
(a) If bi = = 1, (x1, y1) = (((x mod M)+1), y);
(b) If bi = =0, (x1, y1) = (x, ((y mod N)+1)).
Where bi represents the current working bit of I(x,y) and i ≠ 0.

6: Collision detection and insertion of secret bits: If (x1, y1) is the selected position for insertion obtained in step 3 and sj is a secret bit, check whether I(x1, y1) is collision free by examining value of f(x1, y1). If f(x1, y1) is 0, then I(x1, y1) is collision free and embed si at LSB of I(x1, y1). On the other hand if f(x1, y1) is 1, perform a linear probe using equation (11) on the image matrix I starting from (x1, y1) till (x2, y2) for which f(x2, y2) is 0 and embed si into the LSB of I(x2, y2) as described below:

Case 1: $f(x_1, y_1) = = 0$

$I'(x_1, y_1) = (b_7b_6b_5b_4b_3b_2b_1s_j)$ and $f(x_1, y_1) = =1$.

Case 2: f(x1, y1) = = 1

New position is computed using linear probing as

$$(x_{new}, y_{new}) = \begin{cases} (x_1, (y_1 + 1)) \text{ify} < N \\ ((x_1 \text{mod} M) + 1, (y_1 \text{mod} N) + 1) \text{ify} == N \end{cases} \dots (11)$$

till $(x_{new}, y_{new}) = (x_2, y_2)$ for which f(x2, y2) = 0 and $I'(x_2, y_2) = ((b_7b_6b_5b_4b_3b_2b_1s_j)$ and $f(x_2, y_2) = 1$.

7: End

Let us apply the embedding algorithm on a 4 × 4 image matrix I given below:

I =

105	115	112	94
109	115	152	107
121	131	119	6
138	114	22	37

After converting all pixels of the above image matrix into even numbers, the new image matrix

I =

104	114	112	94
108	114	152	106
120	130	118	6
138	114	22	36

and a flag matrix f of size 4 × 4 is defined and initialized to 0 as

f =

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

Let the secret message to be embedded be **1111010100101011**. The message embedding is started from the location (1, 1) whose pixel value is 104. The binary bit Sequence of 104 is 01101000. Now traverse the matrix according to the bit sequence till the 2nd LSB is encountered using Equations (8) and (9) as:

0 1 1 0 1 0 0

I(1,1)→I(1, 2)→I(2, 2)→I(3, 2)→I(3, 3)→I(4, 3)→I(4, 4)→I(4, 1).

After reaching to the position (4, 1), the same position of flag matrix f is examined and found that f(4, 1) is 0, i.e. I(4, 1) is collision free. Thus embed the 1st message bit (here 1) in I(4, 1) which change the pixel value 138 to 139. After embedding f(4, 1) is also changed to 1 as given below

I =

104	114	112	94
108	114	152	106
120	130	118	6
139	114	22	36

and f =

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

This process continues to embed all secret bits and at the same time the value at the corresponding position in the flag matrix is also changed to 1. Now let us give an example to detect collision and resolve it using flag matrix. For this purpose choose the last message bit. After the insertion of the 15 bits, the flag matrix looks like as

f =

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

To insert the last message bit (16th bit) into the cover image, the traversal will be started from the pixel 36 of position (4, 4) into the cover image. According to the bit pattern (first 7 bits) of 36 (here 0010010), the traversal will be stopped in the position (2, 2) and collision occurs as in (2, 2) location of the flag matrix is set as 1. To find the collision free location, apply linear probing in row major order by Equation (11) as

1 1 1 1 0

f(2, 2)→ f(2, 3)→ f(2, 4)→ f(3, 1)→ f(3, 2)→ Stop

As (3, 2) location of flag was set to 0, embed last secret bit (here 1) in the cover pixel 130 of location (3, 2) as given in the last row of Table 1. After embedding the secret bit sequence using above algorithm, the stego-image I' and flag matrix f is

I' =

105	114	112	94
109	115	153	106
121	131	119	6
139	115	23	36

and f=

1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1

Table 1: Example details of the embedding procedure

Start Coordinate (x,y)	I(x,y)	$b_7b_6b_5b_4b_3b_2b_1b_0$	Final coordinate (x_1,y_1)	$I(x_1, y_1)$	s_i	$I'(x_1,y_1)$
(1,1)	104	01101000	(4,1)	138	1	139
(1,2)	114	01110010	(1,1)	104	1	105
(1,3)	112	01110000	(4,3)	22	1	23
(1,4)	94	01011110	(2,2)	114	1	115
(2,1)	108	01101100	(2,4)	106	0	106
(2,2)	114	01110010	(2,1)	108	1	109
(2,3)	152	10011000	(1,3)	112	0	112
(2,4)	106	01101010	(2,3)	152	1	153
(3,1)	120	01111000	(3,4)	6	0	6
(3, 2)	130	10000010	(1,4)	94	0	94
(3,3)	118	01110110	(4,2)	114	1	115
(3,4)	6	00000110	(1,2)	114	0	114
(4,1)	138	10001010	(3,1)	120	1	121
(4,2)	114	01110010	(4,4)	36	0	36
(4,3)	22	00010110	(3,3)	118	1	119
(4,4)	36	00100100	(3, 2)	130	1	131

3.1 Retrieval Technique

The message retrieval is just the reverse procedure of the embedding process as given in the extraction algorithm.

Algorithm 2:Extraction of secret bits from Stego-imageI'

Input: Stego-image I'

Output: Secret bit stream S

1: Begin

2: Binary pattern generation: Obtain binary value of $I'(x,y)$ as $I'(x,y)=(b_7b_6b_5b_4b_3b_2b_1s_i)$ where $b_i \in \{0,1\}$

3: Declaration and Initialization of Flag matrix: Declare a flag matrix F of size M×N and set its all elements to 0.

4: Image matrix traversal: Traverse the image matrix I according to the binary bit pattern of pixel $I'(x,y)$ starting from location (x,y) till the 2nd least significant bit (Here b_1) using the following rules:

(a) If $b_i == 1$, $(x_1, y_1) = ((x + 1) \bmod M, y)$

(b) If $b_i == 0$, $(x_1, y_1) = (x, (y + 1) \bmod N)$

Where b_i represents the current working bit of $I'(x,y)$ and $i \neq 0$

5: Collision detection and extraction of secret bits: If (x_1, y_1) is the selected in step 3, check whether $I'(x_1, y_1)$ is collision free by examining value of $f(x_1, y_1)$. If $f(x_1, y_1)$ is 0, then $I'(x_1, y_1)$ is collision free and extract the LSB s_i of $I'(x_1, y_1)$ and is added into secret bit stream. On the other hand if $f(x_1, y_1)$ is 1, perform a linear probe using Equation (11) on the image matrix I' starting from (x_1, y_1) till (x_2, y_2) for which $f(x_2, y_2)$ is 0 and embed s_i into the LSB of $I'(x_2, y_2)$ as described below:

case 1: $f(x_1, y_1) = = 0$

extract the LSB s_i from $I'(x_1, y_1) = (b_7b_6b_5b_4b_3b_2b_1s_j)$ and set $f(x_1, y_1) = 1$

case 2: $f(x_1, y_1) = = 1$

Linear probing is applied using Equation (11) till $(x_{new}, y_{new}) = (x_2, y_2)$ for which $f(x_2, y_2) = = 0$ and extract the LSB s_i from $I'(x_2, y_2) = ((b_7b_6b_5b_4b_3b_2b_1s_j)$ and set $f(x_2, y_2) = 1$

6: End

To illustrate message retrieval technique first define a flag matrix f of size 4 × 4 as

f =

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

Traversal will start from location (1, 1) in stego-image with the pixel 104, whose corresponding Binary Bit Sequence = 01101001. From the binary sequence of 105 it is observed that first seven bits are unchanged. Only the last bit is changed in worst case (if s_j and b_0 are not matched). Thus traversed path remain same as to the traversed path during embedding. Now traverse the matrix according to the bit sequence till the 2nd LSB is encountered using Equations (8) and (9) as:

0 1 1 0 1 0 0

I(1, 1)→I(1, 2)→I(2, 2)→I(3, 2)→I(3, 3)→I(4, 3)→I(4, 4)→I(4, 1)

After reaching to the position (4, 1), the same position of flag matrix f is examined and found that f(4, 1) is 0, i.e. I(4, 1) is collision free. Thus extract the 1st message bit (here 1) from the LSB of $I'(4, 1)$ whose value is 139. After extracting secret bit s_j from the location (4, 1) of I', f(4, 1) is changed to 1 as given below

f =

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

This process continues to extract the secret bits and at the same time the corresponding bit in the flag matrix is also changed. Now let us give an example to detect collision and resolve it using flag matrix during extraction. For this purpose choose the last bit. After the insertion of the 15 bits, the flag matrix looks like as

f =

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

To extract the last secret bit(16th bit) from the cover image, the traversal will be started from the pixel 36 of position (4, 4) into the cover image. According to the bit pattern (first 7 bits) of 36 (here 0010010), the traversal will be stopped in the position (2, 2) and collision is found as in location (2, 2) in the flag matrix is set as 1. To find the collision free location, apply linear probing by Equation (11) as

1 1 1 1 0

f(2, 2)→ f(2, 3)→ f(2, 4)→ f(3, 1)→ f(3, 2)→ Stop

As location (3, 2) of the flag matrix is set to 0, extract the secret bit (here 1) from the LSB of the pixel 131 at location (3, 2) in stego-image I'as given in the last row of

Table 2. After extracting the secret bit sequence using above algorithm flag matrix f is

f =

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

Table 2: Example details of extracting procedure

Start Coordinate (x,y)	I' (x,y)	$b_7b_6b_5b_4b_3b_2b_1b_0$	Final coordinate (x_1,y_1)	I' (x_1,y_1)	Extracted bits (s_i)
(1,1)	104	01101000	(4,1)	139	1
(1,2)	114	01110010	(1,1)	105	1
(1,3)	112	01110000	(4,3)	23	1
(1,4)	94	01011110	(2,2)	115	1
(2,1)	109	01101100	(2,4)	106	0
(2,2)	115	01110010	(2,1)	109	1
(2,3)	153	10011000	(1,3)	112	0
(2,4)	106	01101010	((2,3)	153	1
(3,1)	121	01111000	(3,4)	6	0
(3, 2)	131	10000010	(1,4)	94	0
(3,3)	119	01110110	(4,2)	115	1
(3,4)	6	00000110	(1,2)	114	0
(4,1)	139	10001010	(3,1)	121	1
(4,2)	115	01110010	(4,4)	36	0
(4,3)	23	00010110	(3,3)	119	1
(4,4)	36	00100100	(3, 2)	131	1

3.2 Security Analysis

In this section, we will explain

Mask= [0 1 1 0];

r_1 = no. of regular groups for M;

Negative mask= [0 -1 -1 0];

r_2 = no. of regular groups for (–M).

Here we assume, N=4, initially set $r_i = s_i = 0$, $1 <= i <= 2$ s_1 = no. of singular groups for M:

s_2 = no. of regular groups for (–M)

$$I' = \begin{bmatrix} 105 & 114 & 112 & 94 \\ 109 & 115 & 153 & 106 \\ 121 & 131 & 119 & 6 \\ 139 & 115 & 23 & 36 \end{bmatrix}$$

For total group/2:
In iteration-1, G_c={105,114,112,94} and $f(G_C)$ = 29;
$F_M(G_C)$ = {105,115,113,94} and $f(F_M(G_C))$=31.
Therefor $f(F_M(G_C)) > f(G_C)$;
Hence, **Regular Group** set $\mathbf{r_1=1.}$
Again $F_{-M}(G_c)$= {105,113,111,94} and $f(F_{-M}(G_c))$ = 27.
Therefore $f(F_{-M}(G_c)) < f(G_C)$
Hence, **Singular Group** set $\mathbf{s_2 = 1.}$

In iteration 2, G_c={109,115,153,106} and $f(G_C)$ = 91,
$F_M(G_C)$ = {109,114,152,106}, and $f(F_M(G_C))$ = 89.
Therefore $f(F_M(G_C)) < f(G_C)$.
Hence, **Singular Group** set $\mathbf{s_1 = 1.}$
Again $F_{-M}(G_c)$ = {109,116,154,106} and $f(F_{-M}(G_c))$ = 93.
Therefore $f(F_{-M}(G_c)) > f(G_C)$.
Hence, **Regular Group**set $\mathbf{r_2=1.}$

RM = r_1/total group = 1/2;
RM' = r_2/total group = 1/2;
SM = s_1/total group = 1/2;
SM'= s_2/total group = 1/2;
d_0 = (RM – SM) = (1/2 – 1/2) = 0;
d_{-0}= (RM' – SM') = (1/2 – 1/2) = 0.

For total group:-
From Iteration 1, $r_1 = 1, s_2 = 1$;
From Iteration 2, $s_1 = 1$, $r_2 = 1$;
In iteration 3, G_c={121,131,119, 6} and $f(G_C)$ = 135,
$F_M(G_C)$ = {121,130,118,6}, and $f(F_M(G_C))$ = 133.
Therefore $f(F_M(G_C)) < f(G_C)$.
Hence, **Singular Group** set $\mathbf{s_1 = 2}$.
Again $F_{-M}(G_c)$ = {121,132,120,6} and $f(F_{-M}(G_c))$ = 137.
Therefore $f(F_{-M}(G_c)) > f(G_C)$.
Hence, **Regular Group**set $\mathbf{r_2=2.}$

In iteration 4, G_c={139,115,23,36} and $f(G_C)$ = 129,
$F_M(G_C)$ = {139,114,22,36}, and $f(F_M(G_C))$ = 131.
Therefore $f(F_M(G_C)) > f(G_C)$.
Hence, **Regular Group**set $\mathbf{r_1=2.}$
Again $F_{-M}(G_c)$ = {1391162436} and $f(F_{-M}(G_c))$ = 127.
Therefore $f(F_{-M}(G_c)) < f(G_C)$.
Hence, **Singular Group**set $\mathbf{s_2 = 2.}$

RM = r_1/total group = 2/4 = 1/2;
RM' = r_2/total group = 2/4=1/2;
SM = s_1/total group = 2/4 = 1/2;
SM'= s_2/total group = 2/4 = 1/2;
d_1 = (RM – SM) = (1/2 – 1/2) = 0;
d_{-1}= (RM' – SM') = (1/2 – 1/2) = 0.

$$2(d_1 + d_0)x^2 + (d_{-0} - d_{-1} - d_1 - 3d_0)x + d_0 - d_{-0} = 0$$
$$2(0+0)x^2 + (0+0-0-3*0)x + 0 - 0 = 0.$$

$$\boldsymbol{x_1 = x_2 = 0.}$$

Hence Probability of embedding,

$$\boldsymbol{P = x/(x - 0.5) = 0/(0 - 0.5) = 0.}$$

4 Experimental Results

Huffman encoding is a lossless compression technique, which can also encode message and produce binary bit stream. Secret data are encoded and compressed by Huffman encoding and secret bit sequence. Table 3 shows 521×512 grayscale 10 cover images: Airplane, Baboon, Barbara, Boat, Couple, Goldhill, Lena, Man, Peppers and Stream. The Peak Signal to Noise Ratio (PSNR) is applied to compare visual quality between the cover image and stego-image. The definition of PSNR is given below

$$PSNR(dB) = 20\log_{10}\frac{255}{\sqrt{MSE}} \dots\dots\dots (12)$$

MSE is the mean squared error between the original image and the modified image which is defined as

$$MSE = \frac{1}{M \times N}\sum_{x=1}^{M}\sum_{y=1}^{N}\left(I(x,y) - I'(x,y)\right)^2 \ldots\ldots\ldots \quad (13)$$

where M and N denotes the width and height of the cover and stego image respectively. Table 3 shows the PSNR values of LSB, RHTF based LSB and our scheme for 90% embedding. In Table 3, we see that the average PSNR is better than LSB and RHTF method.

Table 3: PSNR result for embedding same message

Cover-images (512×512)	LSB	RHTF based LSB	Our method
	PSNR (dB)	PSNR (dB)	PSNR (dB)
Airplane	50.2441	50.2273	50.2938
Baboon	49.9268	49.9983	50.2771
Barbara	50.7814	50.1712	50.8807
Boat	51.1182	51.1892	51.2400
Couple	51.1192	51.1715	51.0477
Goldhill	50.4200	50.5482	50.4287
Lena	51.1189	50.7136	50.7930
Man	50.6599	50.3456	50.8573
Peppers	50.2590	50.2675	50.2300
Stream	50.9532	51.0419	51.4591
Average	50.6600	50.5674	50.7513

4.1 Statistical Attack

In this section, we employ two statistical attacks to evaluate the security of our proposed method, LSB method and RHTF based LSB.

4.1.1 Resisting the Chi Square Attack

We have already discussed that our proposed algorithm is completely based on random embedding of messages on the LSB plane of the cover image. The most interesting thing while resisting the Chi Square is that Chi Square is completely unable to detect the existence of message if embedding is done at a random basis. So it fails to detect the message embedded by our proposed algorithm. After applying the Chi Square Attack on the stego image of Lena obtained by our algorithm we have got the graph shown in Figure 1(b) which proves the inability of detection by Chi Square Steganalysis Technique. On the other hand from Figure 1(a), it is clearly observed that Chi Square Steganalysis technique is very effective at detecting the stego-images using LSB method.

Figures 1 and 2 show a comparison among LSB and our scheme in terms of resistance performance against Chi Square and RS attacks. From the figure it is clear that, any stego-image generated by LSB scheme is detectable by this attack i.e. no stego-image is identified as cover image for 90% embedding. But RHTF based LSB is secured against Chi Square. Figure 1 shows that even for the 90% embedding, 90.52% stego-images generated by RHTF based LSB are identified as cover image when Chi Square attack is applied. On the other hand, our proposed method can produce 96.3% stego-images as cover image for 90% embedding. Accordingly, we can say that our proposed scheme is more secured than LSB and RHTF based LSB against this attack.

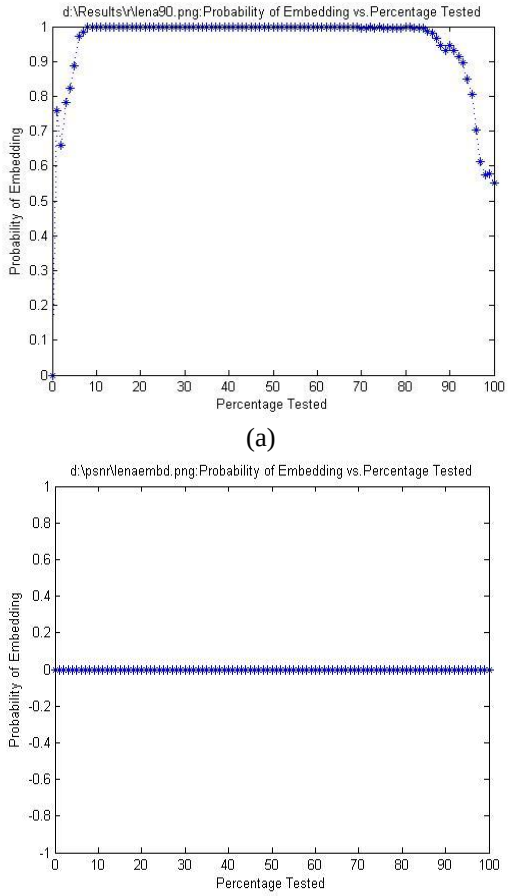

Figure 1: Results of Chi Square Attack on stego image obtained by applying (a) LSB technique and (b) Proposed algorithm with 90% embedding

4.1.2 RS-Attack

Figure 2 shows RS-attack result of gray stego-Lena image with size of 512×512 using our proposed method. The X-axis represents the percentage of embedding and Y-axis represents relative percentage of the regular and singular groups with masks M =[0 1 1 0] and –M =[0 -1 -1 0]. Figure 2 shows that expected value of R_m is almost equal to the value of R_{-m} and S_m is equal to S_{-m}. The other stego-images generated by our proposed method are also tested and produce same results as what Figure 2 represents. Thus we can conclude that our proposed method is secured against the RS-steganalysis.

5 Conclusions

In this paper, we propose a novel technique to produce a better stego-image that is secured against both χ^2 attack and RS attack compared to LSB substitution method and RHTF based LSB approach due to location selection using linear probing. Besides the experimental results shown that the

produced stego-image is totally indistinguishable from the cover image by human eye.

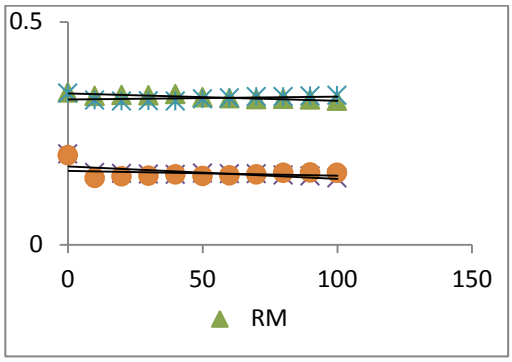

Figure 2: RS diagram of graystego-Lena image with size of 512×512 using our proposed method, where X-axis represents percentage(%) of Embedding and Y-axis represents relative number of Regular and Singular Groups.

References

[1] National Bureau of Standard (U.S), *Data Encryption Standard (DES)*, Federal Information Processing Technical Information Service, Springfield, VA, 1997.

[2] C.C. Chang and H. W. Tseng, "A steganographic method for digital images using side match," *Pattern Recognition Letters*, pp. 1431-1437, 2004.

[3] A. Chedded, J. Condell, K. Curran, and P. M. Kevitt, "Digital image steganography: Survey and analysis of current methods," *Signal Processing*, vol. 90, pp. 727-752, 2010.

[4] J. Fridrich, M. Goljan, and R. Du, "Reliable detection of LSB steganography in color and grayscale images," in *Proceedings ACM Workshop Multimedia and Security*, pp. 27-30, 2001.

[5] A. A. Hanafy, G. I. Salama, and Y. Z. Mohasseb, "A secure covert communication model based on video steganography," in *Proceedings of the 2008 IEEE Military Communications Conference*, pp. 1–6, 2008.

[6] T. Y. Liu and W. H. Tsai, "A new steganographic method for data hiding in Microsoft Word documents by a change tracking technique," *IEEE Transaction on Information Forensics and Security*, vol. 2 no. 1, pp. 24-30, 2007.

[7] H. Luo, F. X. Yu, H. Chen, Z. L. Huang, H. Li, and P. H. Wang, "Reversible data hiding based on block median preservation," *Information Science*, vol. 181, no. 2, pp. 308-328, 2011.

[8] W. Luo, F. Huang, and J. Huang, "Edge adaptive image steganography based on LSB matching revisited," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 2, pp. 201-214, 2010.

[9] D. C. Lou and C. H. Hu, "LSB steganographic method based on reversible histogram transformation function for resisting statistical steganalysis," *Information Science*, doi:10.1016/j.ins.2011.06.003, 2011.

[10] B. Karthikeyan, S. Ramakrishnan, V. Vaithiyanathan, S. Sruti, and M. Gomathymeenakshi, **"**An improved steganographic technique using LSB replacement on a scanned path image**,"** *International Journal of Network Security*, vol. 15, no. 1, pp. 314-318, Jan. 2013

[11] A. R. S. Marçal and P. R. Pereira, "A steganographic method for digital images robust to RS steganalysis," in *International Conference on Image Analysis and Recognition*, LNCS 3656, pp. 1192-1199, 2005.

[12] J. Mielikainen, "LSB matching revisited," *IEEE Signal Processing Letters*, vol. 13, no. 5, pp. 285-287, 2006.

[13] A. Nissar and A. H. Mir, "Classification of steganalysis techniques: A study," *Digital Signal Processing*, vol. 20, pp. 1758-1770, 2010.

[14] R. Petrovic, J. M. Winograd, K. Jemili, and E. Metois, "Data hiding within audio signals," in *Proceedings of the 4th International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Services*, pp. 88-95, 1999.

[15] R. Rivest, A. Shamir, and Adleman, "A method for obtaining digital signetures and public-key cryptosystems," *Communication of ACM*, vol. 120-126, 1978.

[16] P. Shah, P. Choudhari, and S. Sivaraman, "Adaptive wavelet packet based audio steganography using data history," in *Proceedings of the 2008 IEEE Region 10 and the 3rd International Conference on Industrial and Information Systems*, pp. 1-5, 2008.

[17] M. H. Shirali-Shahreza and M. Shirali-Shahreza, "A new approach to Persian/Arbic text steganography," in *Proceedings of Fifth IEEE/ACIS International Conference on Computer and Information Science*, pp. 310-315, 2006.

[18] W. Stallings, *Cryptography and Network Security – Principles and Practice*, 4th ed. Pearson Education Pvt. Ltd., Indian, 2004.

[19] B. Wang and J. Feng, "A chaos-based steganography algorithm for H.264 standard video sequences," in *Proceedings of the 2008 International Conference Communications, Circuits and Systems*, pp. 750-753, 2008.

[20] H. Wang and S. Wang, "Cyber warfare: steganography vs. steganalysis," *Communication of the ACM*, vol. 47, no. 10, pp. 76-82, 2004.

[21] A. Westfeld and A. Pfitzmann, "Attacks on steganographic systems," in *Proceedings of the 3rd International Workshop on Information Hiding*, Dresden,Germany, pp. 61-76, 1999.

Amitava Nag obtained his M.Tech from University of Calcutta in the year 2005. He earned his B.Techfrom Dept. of Engineering & Technological Studies, University of Kalyani in the year 2003. He is presently working as an Assistant Professor in Academy of Technology, India and

also working towards his PhD at the Dept. of Engineering & Technological Studies, University of Kalyani. He is a member of IEEE and CSI,India.His area of interest includes Cryptography and steganography.

S. Biswas obtained his Ph.D in engineering from Jadavpur University in the year 2004. He obtained his M.E from Jadavpur University and B.E from Bengal Engineering College (Presently known as Bengal Engineering and Science University, Shibpur) in the year 1994 and 1990 respectively. He is presently working as Scientific Officer (Associate Professor Rank) at the Dept. of Engineering & Technological Studies, University of Kalyani. He has more than 14 years of teaching experience. His area of interest includes, Artificial Neural Network, Image Processing, Frequency Selective Surfaces, Microstrip Antennas.

D. Sarkar has obtained her Ph.D degree in Engineering from Jadavpur University in the year 2005. She has obtained her M.E and B.E from Bengal Engineering College (presently known as Bengal Engineering and Science University, Shibpur) in the year 1994 and 1991 respectively. She is presently working as Scientific Officer (Associate Professor Rank) at Dept. of Engineering & Technological Studies, University of Kalyani. She has more than 14 years of teaching experience. Her area of research includes Artificial Neural Network, Microstrip Antenna, Frequency Selective Surfaces, and Embedded Systems.

Partha Pratim Sarkar obtained his Ph.D in engineering from Jadavpur University in the year 2002. He has obtained his M.E from Jadavpur University in the year 1994. He earned his B.E degree in Electronics and Telecommunication Engineering from Bengal Engineering College (Presently known as Bengal Engineering and Science University, Shibpur) in the year 1991. He is presently working as Senior Scientific Officer (Professor Rank) at the Dept. of Engineering & Technological Studies, University of Kalyani. His area of research includes, Microstrip Antenna, Microstrip Filter, Frequency Selective Surfaces, and Artificial Neural Network. He has contributed to numerous research articles in various journals and conferences of repute. He is also a life Fellow of IETE.