ijecs open access www.ijecs.in

International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 3 Issue 6 June, 2014 Page No. 6770-6778

Data Hiding using difference between adjacent pixels and bit plane swapping

Shalini Singh, Satakshi Sharma

Assistant Professor
Electronics and Communication Engineering
ACME, Maharishi Dayanand University, Haryana, India
Student
Student Electronics and Communication Engineering
ACME, Maharishi Dayanand University, Haryana, India
satakshisharma03@gmail.com

Abstract-In this paper, we have implemented a novel technique for hiding text messages in a gray scale image. This technique involves data hiding using cryptography and steganography, which makes it a strong data hiding mechanism. In this method, we first apply cryptography to secret message and then hide it inside a gray scale image using steganography.

Keywords-Cryptography; Steganography; Data hiding, Bit Plane swapping

I. INTRODUCTION

The importance of data hiding is well known. Nowadays, as the geographical distances between friends and family has increased, along with it has increased the amount of sharing of information through common channels. Besides friends and family, information is exchanged between organizations. Many a times, the information shared is of critical importance and should be shared secretly. When information is shared, it is important to ensure that the shared information reaches the intended recipient safely, without any discrepancy. It is also important, that information is not made available to unintended recipients. Thus, when the information/data is shared between two parties, it is important that:

- Data should reach the intended recipient, unaltered.
- The data should not reach tampered.
- The data should not fall in the hands of unintended recipients
- Even if data lands with unintended recipients, they should not be able to find the relevant information carried by the data. In other words, information should be made secure.

With the rising amount of sharing of important and secret data between organizations and for making transfers secure, there has come up a need for efficient data hiding techniques.

An efficient data hiding technique should posses the following features:

- It should enable the sender to hide and intended recipient to unhide information easily.
- It should not let the hidden secret data get tampered by external attacks i.e. received data should be same as the sent data.
- It should not reveal the secret information to any recipient who is trying to access the secret data without knowing the key.

In this paper, we have implemented data hiding with the help of steganography. In order to make the technique more secure, we added another level of security by applying cryptography to the secret data before it undergoes steganography.

II. DATA HIDING TECHNIQUES

For secure transmission, we can hide secret data using several ways. The data hiding technique is chosen keeping in mind the type of secret data. Secret data can either be text, image, audio or video. Depending upon the type of data, appropriate data hiding mechanism can be selected.

There are many data hiding techniques. Some turn the data into non-understandable format, whereas some hide the existence of secret data. Few of the data hiding mechanisms are [1]:

- Water Marking
- Cryptography
- Steganography
- Fingerprinting
- Digital Signature

Here, we will concentrate on cryptography and steganography as they form the base of our work presented in this paper. In the following sections, we first describe steganography and cryptography and then explain the difference between the two. Then we introduce the cryptographic and steganographic algorithms we have used to encrypt the secret data and to hide the data inside an image respectively.

A. CRYPTOGRAPHY

Cryptography is the art and science of devising methods that allow information to be sent in a secure form in such a way that the only person able to retrieve this information is the intended recipient. [2]

Cryptography involves two major processes – encryption and decryption. The process of conversion of plain text to cipher text is known as encryption whereas the process of conversion of cipher text back to plain text is known as decryption.

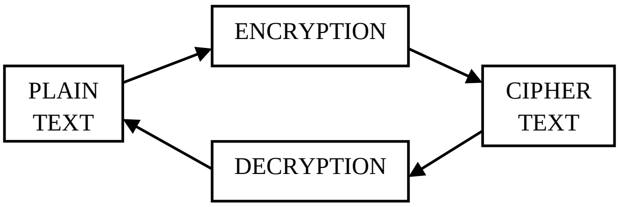

Figure 1: Process of Cryptography

In cryptography, users (sender and the receiver) decide on an algorithm to be adopted. This algorithm is then used to first encrypt the secret data at the sender's end and then followed by the recipient to decrypt the encrypted data in order to reveal the original secret message.

For example: Ceaser's Cipher Algorithm

CEASER'S CIPHER ALGORITHM

Sender: Shifts each character of the message with +3 places and sends the encrypted message to the receiver

Secret message	A	T	T	A	C	K
Encrypted message	D	W	W	D	F	N

Receiver: Shifts character of the encrypted message with -3 places and thus obtains the original secret message

Encrypted message	D	W	W	D	F	N
Secret message	A	T	T	A	C	K

Figure 2: An example of Ceaser's Cipher Algorithm

B. STEGANOGRAPHY

Steganography is the art of hiding data in such a way that the existence of the secret data is known only to the involved parties.

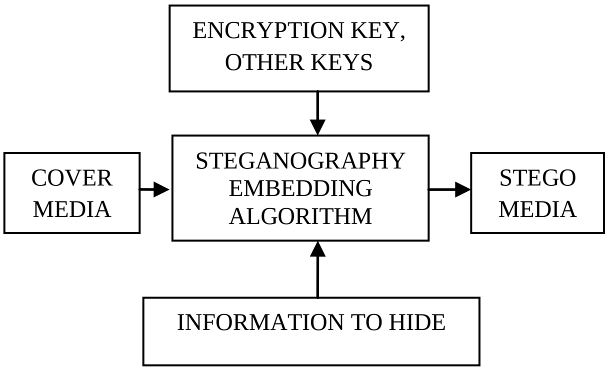

Figure 3: Information Hiding Process

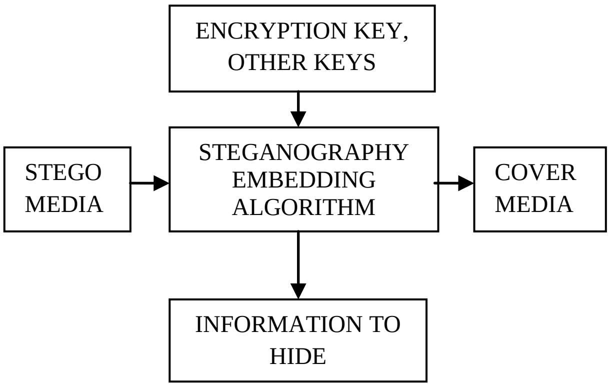

Figure 4: Information Extraction Process

In steganography, information is hidden in a medium called cover medium in such a way that no one apart from the sender and the intended recipient are aware about it. So, it does not attract unwanted attention of the attackers.

C. DIFFERENCE BETWEEN CRYPTOGRAPHY AND STEGANOGRAPHY

In cryptography, the secret data is made non-understandable by application of algorithms whereas in steganography, the secret data is hidden inside a cover medium in such a way that its existence is hidden. As a result, information hidden using cryptography attracts more unwanted attention than information hidden using steganography.

Both, cryptography and steganography are very efficient techniques for data hiding and secure sharing. Thus, when used in combination, these two techniques highly enhance the security of the data.

In this paper we have used both cryptography and steganography to device a novel technique for data hiding.

III. STEGANOGRAPHY – RECENT DEVELOPMENTS

Most prominent work in the field of steganography came up by Marvel, in 1999 [3]. Since then, this field is witnessing only evolution. Steganography can either be reversible or irreversible. In an irreversible steganographic technique, once the data is hidden within the cover image, on extraction, only the hidden data will be extracted. The cover image cannot be reconstructed in case of irreversible steganography. On the other hand, with reversible steganographic techniques, on extraction of the hidden data, we can get the data as well as the reconstructed image.

In medical and military applications there are instances where the cover image is very crucial and even slightest of distortion is intolerable. In such cases, it becomes very important to implement reversible steganography.

Recently, many reversible steganographic techniques have been introduced [4, 5]. The scheme of Ni et al finds out all the pixel values of the cover image and plots a histogram with the values obtained [6]. In this scheme, the peak points and zero points are selected and the pixel values are shifted between each point-pair, thus shifting the histogram and making peak point available for hiding the message bits. It should be noted that in this method, there is a limitation. The maximum number of bits that can be hidden inside the cover image is the number of pixels of peak value. Hence, though this algorithm gives a high quality stego-image, its hiding capacity must be improved.

With an aim of increasing the hiding capacity, a new technique called 'Adjacent pixel difference' (APD) was introduced by Li et al [7]. Where, in the scheme proposed by Ni [6], the histogram is plotted for all the pixel values; in APD, the histogram is plotted for the values obtained after finding the difference of neighboring pixels. The property of an image that this scheme exploits is 'smoothness'. For almost all the images, the value of neighboring pixels does not change suddenly. The difference of neighboring pixels is usually a small value, thus increasing the frequency of peak points. Hence APD gives higher hiding capacity than the other data hiding scheme [6].

Here in this paper, we have employed APD (without PRNG) for implementing steganography, i.e. for hiding secret message inside the cover image.

IV. CRYPTOGRAPHY – BIT PLANE SWAPPING ALGORITHM

In APD with bit plane swapping, bit planes are swapped before the message undergoes steganography.

The scheme for bit plane swapping is as follows:

1. The secret message to be hidden is entered by the user
2. All the entered characters are converted into ASCII and then to binary.
3. The binary value of ASCII characters comprises of 7 bits and hence 7 different bit planes from LSB, LSB+1, LSB+2,…., MSB.
4. The sender selects the bit planes he wishes to swap. Suppose, the sender selects LSB bit plane to be swapped with LSB+3 and LSB+1 bit plane to be swapped with MSB, then the LSB bit of each character will be replaced with its LSB+3 bit and vice versa. Also the LSB+1 bit of each character will be replaced with its MSB bit and vice versa. In this way, the bit plane swapping will occur.

The major advantage of bit plane swapping method is that, even if the message is as small as a single character, it will be encrypted and cannot be decrypted unless the key is known.

Suppose the secret message to be hidden is:	this
ASCII values	t – 116 h – 104 i – 105 s – 115
Binary values	t - 1 1 1 0 1 0 0 h - 1 1 0 1 0 0 0 i - 1 1 0 1 0 0 1 s - 1 1 1 0 0 1 1
Bit planes decided	MSB swapped with LSB+4

to be swapped (depends on the sender)	LSB+5 swapped with LSB+3 LSB+2 swapped with LSB

Binary values after bit plane swapping	1 0 1 1 0 0 1
	0 1 1 1 0 0 0
	0 1 1 1 1 0 0
	1 0 1 1 1 1 0

Message to be sent for hiding using steganography	1011001011100001111001011110 (Concatenation of all rows)
ASCII values after swapping	89 – Y 56 – 8 60 – < 94 – ^
Encrypted message	Y8<^

Figure 5: Bit plane swapping – Illustration

After bit plane swapping, the encrypted message is sent for hiding within the cover image. The encrypted message is hidden using APD (without PRNG concept).

V. ADJACENT PIXEL DIFFERENCE (APD) ALGORITHM

A natural image has local similarity. Therefore, the difference between adjacent pixels is close to zero and many neighboring pixels have zero difference. This restricts the histogram to mostly around values near zero.

Type I pixel-difference function is employed to calculate the difference between each current pixel and the next pixel. Then, APD generates the histogram of the transformed image, which is called the difference sequence. APD selects the first pair of a peak point and its closest zero point. Given a peak point, both sides of the peak point have a zero point that is closest to the peak point. Let the individual values of first peak point, the zero point closest to its left, and the zero point closest to its right be PP_1, ZPL_1, and ZPR_1, respectively, where, $ZPL_1 < PP_1 < ZPR_1$. The closest zero point to the first peak point is denoted as CZP_1. Eq. (1) is the selection function.

The selection avoids the second peak point, PP_2, in order to fall in the range between the first pair's points.

$$CZP_1 = \begin{cases} ZPL_1, if\ PP_2 \notin [ZPL_1, PP_1], \\ ZPR_1, otherwise \end{cases} \quad (1)$$

According to the pair of the first peak point and the closest zero point, APD shifts the histogram from PP_1 to CZP_1 by increasing one (or decreasing one) and then embeds data. To improve the hiding capacity, APD can select the second pair of the peak point and the closest point to shift. Let the individual values of the second peak point, the zero point closest to its left, and the zero point closest to its right be PP_2, ZPL_2, and ZPR_2, respectively, where $ZPL_2 < PP_2 < ZPR_2$. APD uses Eq. (2) to select the closest zero point, CZP_2, of the second peak point. The intersection between the ranges of the two pairs is empty.

$$CZP_2 = \begin{cases} ZPL_2, if\ PP_2 \notin [ZPL_2, PP_2]\ and\ CZP\ _1 \notin [ZPL_2, PP_2] \\ ZPR_2, otherwise \end{cases} \quad (2)$$

When two or more than two peak points have the same peak value, APD selects the leftmost peak point for the first peak point and selects the rightmost peak point for the second peak point. Finally, APD runs the reverse pixel-difference transformation (Type II) to obtain a high quality stego-image. In the data extraction process, APD scans the stego-image to extract the hidden data and recover the original pixels in order. In order to enhance the security of the hidden data, APD uses a pseudo-random number generator (PRNG). The seed of PRNG must be sent to the receiver for data extraction. The security of the algorithm is dependent on the length of the seed. Larger the length of the seed, more secure is the APD algorithm. If, a smaller seed is chosen, it will directly affect the security of the algorithm.

Thus, to make the security of the algorithm independent of the varying length of the seed, we have eliminated the concept of PRNG and introduced cryptography via bit plane swapping of the secret message.

A. DATA EMBEDDING AND DATA EXTRACTION PROCESS

The scheme of APD is as follows:

(1). Data embedding process

Step 1. *Type I transformation*

APD scans the cover image in inverse s-order, row by row, top to bottom. Then, APD employs the Type I pixel-difference function to generate a one-dimensional difference sequence. Let P_i be the pixel value with index value i in the scanned sequence, and let the total number of pixels be n. The pixel-difference function of Type I is as follows:

$$P'_i = \begin{cases} P_i, if\ i = 0, \\ P_{i-1}, if\ 1 \le i \le n-1, \end{cases} \quad (3)$$

where P'_i is the transformed pixel value. The value P'_i is called the original difference coefficient. This step transforms an image into an APD coefficient sequence called the Type I pixel-difference transformation.

Figure 6: APD with bit plan

Step 2. *Select pairs of peak and zero points*

APD generates the histogram of the coefficient sequence. Then, APD selects the first pair of PP_1 and CZP_1. If the capacity is insufficient, APD selects the second pair, PP_2 and CZP_2. If no zero point exists, APD selects a minimum frequency point. APD stores the extra information of these coefficient values. Then, the minimum frequency point is cleaned to become the zero point.

Step 3. *Shift and embed data*

Let $sd_j \in \{-1, 1\}$ be one of the two shift directions, where $j \in \{1, 2\}$. If $PP_j < CZP_j$, then $sd_j = 1$, else $sd_j = -1$. APD scans the coefficient sequence skipping the first value. APD adds each coefficient value in the range $[PP_j + sd_j, CZP_j]$ by sd_j. Then, once APD finds the coefficient with value PP_j, it checks the next embedded bit. If the embedded bit is "0", then the coefficient value does not change. If the embedded bit is "1", then the coefficient value adds sd_j , becoming $PP_j + sd_j$. This step cleans the point of $PP_j + sd_j$ to hide data.

Step 4. *Type II transformation*

This step executes the Type II pixel-difference transformation to produce a slightly modified stego-image. To obtain ue (P'''_i) of the stego-image sequence, the step recovers the original pixel value by the previous index value (P_{i-1}). If the original image has been deleted, the P_i value also can be easily worked out from the P_{i-1} value, the P''_i value, shift

direction, and information about the peak and zero points. The pixel difference function of Type II is as follows:

$$P'''_i = \begin{cases} P''_i, if\ i = 0, \\ P_{i-1} - P''_i, otherwise \end{cases} \qquad (4)$$

In Eq. (4), using the P_{i-1} value to generate P'''_ican avoid propagating the pixel difference. This step transforms the P'' sequence into a stego-image pixel sequence. To avoid overflow and underflow, APD applies a location map to store the information and to fix the pixel values.

Step 5. *Generate the stego-image and the key information*

Construct the stego-image from the final pixel sequence of Step 4 and generate the parameters, referred to as key information (KI), which must be sent to the receiver for data extraction.

(2). **Data extraction process**

To extract the embedded data and to recover the original pixel values, the APD method scans the stego-image in the same order specified in Step 1 of the data embedding process. The scheme is as follows:

Step 1. *Scan the stego image*

APD scans the stego-image in an inverse s-order, row by row and top to bottom, to generate the sequence P'''.

Step 2. *Inverse Type II transformation*

APD obtains the pairs of peak points and zero points from KI. APD employs the inverse Type II pixel-difference function to generate the one-dimensional difference sequence P''. The transformation includes two sub-problems

Step 2.1. The transformation requires the original pixel values. According to the location map, if $P'''_i = 0$ and underflow, then APD recovers $P'''_i = -1$. If P_i = 255 and overflow, then the recovered $P'''_i = 0$. After APD recovers the P'''_i sequence, the function produces the original pixel sequence, P , as follows:

$$P_i = \begin{cases} P'''_i, \quad if\ i = 0, \\ P'''_i + sd_j, elseif\ 1 \le i \le n-1\ and\ (P_{i-1} - P'''_i) \in \left(PP_j, CZP_j\right], j \in \{1,2\} \\ P'''_i, \quad otherwise \end{cases} \tag{5}$$

APD employs P_{i-1} and P'''_i to obtain the P_i value and to avoid propagating the pixel difference.

Step 2.2. Then, APD produces the P'' sequence employing Eq. (6).

$$P''_i = \begin{cases} P_i, if\ i = 0, \\ P_{i-1} - P''_i, otherwise \end{cases} \tag{6}$$

Step 3. *Produce original pixel sequence and extract the embedded value*

According to the sequence, APD extracts the embedded values as follows:

$$Secret\ bit = \begin{cases} 0\ if\ P''_i = PP_j, \\ 1\ elseif\ P''_i = PP_j + sd_j \end{cases} \tag{7}$$

where j ∈ {1, 2}. (After this step, original APD method applies PRNG to obtain the correct embedded order of data, which we have not done because of introduction of bit plane swapping.) (Refer [7] for APD example)

Step 4. *Cover image reconstruction*

Construct the original image from the P sequence.

B. THE KEY INFORMATION

The key information includes the point-pairs information, the location map information and key to decrypt the data. The location map is used to surmount the underflow and overflow problem. To describe the three parts of the key, the essential symbols and their meanings are listed as follows:

KI: The key information
KD: The key to decrypt
PPI: The information of the point-pairs
LM: The information of location map

In original APD, the length of seed of PRNG dominates the security of hiding data. The sender pre-defines the length of seed and generates seed randomly. If the length of seed is large, then the security will be high but if the length is chosen to be small, then it will disturb the security of the algorithm. Thus, to make the security of the algorithm independent of length of seed, we have eradicated the concept of PRNG and employed bit plane swapping. APD generates KI as follows:

KI = PPI • LM • KD, (8)

where • denotes the operation of concatenation.

In PPI, the information regarding the peak points and zero points are sent. LM contains information regarding the location map which tells about value of boundary pixels and out pixels (described in detail in nest section). KD is the key to decrypt the data. This key is the concatenation of the bit swap order and the message length. E.g. If the message is of the length 70 and if the bit planes are numbered in the following way:

MSB	LSB+5	LSB+4	LSB+3	LSB+2	LSB+1	LSB
1	2	3	4	5	6	7

And the decided bit plane swapping is

MSB swapped with LSB+4
LSB+5 swapped with LSB+3
LSB+2 swapped with LSB

Then, the KD will be:
Swap order • message length = 341276570

C. OVERFLOW AND UNDERFLOW

The shift operation may result in the pixels in the boundaries overstepping the boundaries. To avoid overflow and underflow, APD applies a location map to store the information and to fix the pixel value. A pixel with the boundary value is called the boundary pixel. If a pixel value exceeds the range, [0, 255], the pixel is called an "out pixel". APD deals with the overflow and underflow problem in the two processes as follows:

Data embedding process In Step 4, APD generates the P'''sequence for the stego-image. In the P'''sequence, all boundary pixels and out pixels should be recorded so that we will know if their pixel values are out of the boundary. APD applies a location map to store the information for all boundary pixels and out pixels. If the pixel P'''_i , is a boundary pixel, then the corresponding bit of the location map is "0". If the pixel, P'''_i, is an out pixel, then the corresponding bit of the location map is "1", and the pixel value of P'''_i is fixed to be the boundary value. Let LB_j be a bit of the location map, where j is the index in the location map. APD employs Eq. (9) to deal with the overflow and underflow problem.

$$\left(P'''_i, LB_j, i, j\right) = \begin{cases} (P'''_i,0, i+1, j+1), if\ P'''_i is\ a\ boundary\ pixel \\ (P'''_i + sd_k,1, i+1, j+1), elseif\ P'''_j is\ an\ out\ pixel \\ (P'''_i, LB_j, i+1, j+1)\ otherwise \end{cases} \tag{9}$$

where 0 ≤ j ≤ i ≤ n −1 and k ∈ {1, 2}. According to Steps 3 and 4 of the data embedding process, if $P'''_i = -1$, then $sd_k = 1$ (or if $P'''_i = 256$, then $sd_k = -1$). Thus, APD fixes the out pixel to be the boundary pixel and stores the bit "1" in the location map.

Data extraction process. In Step 2.1, if the LM ≠ ϕ, APD applies the location map and the P'''_i sequence, which does not include the out pixels, to transform the sequence so that it includes the out pixels. Eq. (10) is the transform function.

$$(P'''_i, i, j) = \begin{cases} (P'''_i, i+1, j+1), if\ LB_j = 0\ and\ P'''_i is\ a\ boundary\ pixel \\ (P'''_i + sd_k, i+1, j+1), elseif\ LB_j = 1\ and\ P'''_i is\ a\ boundary\ pixel \\ (P'''_i, i+1, j)\ otherwise \end{cases} \tag{10}$$

where 0 ≤ j ≤ i ≤ n − 1 and k ∈ {1, 2}. According to the location map, if $P'''_i = 0$ and $LB_j = 1$ then $sd_k = 1$. APD recovers $P'''_i = -1$. If $P_i = 255$ and $LB_j =$ "1" then $sd_k = -1$. P'''_i is recovered as 256. After APD obtains the fixed P''' sequence, APD can generate the correct original pixel sequence, P.

VI. RESULTS AND CONCLUSION

When the secret data needs to be exchanged between two parties, the sender first encrypts the data using bit plane swapping. This encrypted data is hidden inside a gray scale cover image using adjacent pixel difference algorithm. This algorithm gives a good amount of data hiding capacity. In case of APD algorithm with PRNG, the security was limited by the length of the seed [4]. In the technique proposed in this paper, we have removed this limitation and added another level of security by making the encryption process independent of length of the seed. If the receiver fails to enter the correct decrypting key, the secret message is not decrypted even after it is extracted from the stego-image. This technique returns the receiver, an encrypted message which has to be decrypted to reveal the original message. This has made the algorithm more secure and also removal of the length limitation has made hiding of even very small messages secure.

Recently, a few methods have been developed to further enhance the hiding capacity of APD algorithm [8]. We are working on introducing modifications to the proposed method in order to provide a better image quality and larger data hiding capacity.

VII. REFERENCES

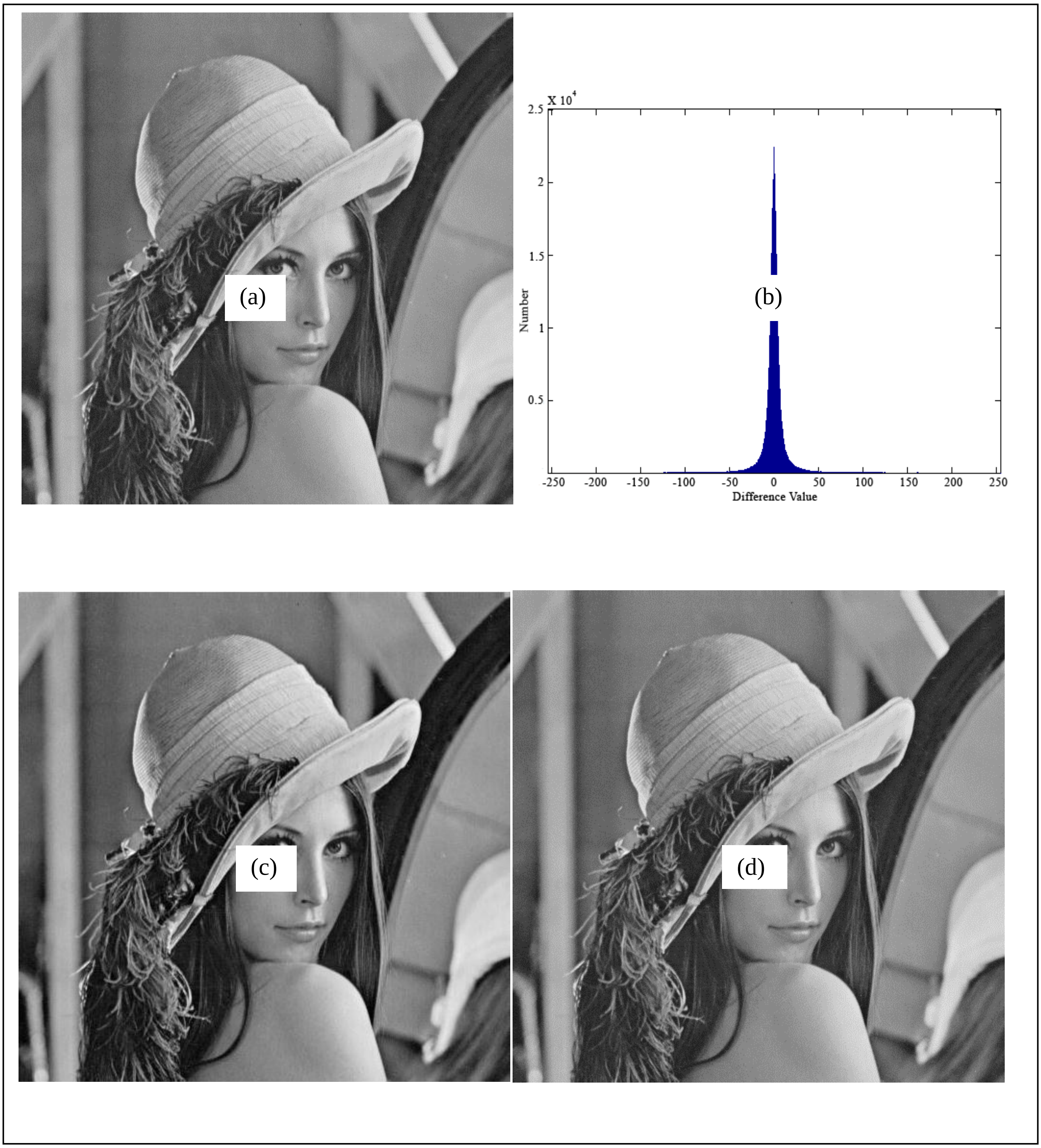

Figure 7: (a) Lena – Cover Image (b) Lena – Histogram (Difference Values)
(c) Lena – Stego Image (d) Lena – Recovered Image

[1]. Dr. K. Sathiyasekar, “A research review on different data hiding techniques”, IJECS vol.3 Issue 1 Jan, 2014, pp. 3655-3659

[2]. Ranjan Bose, “Information Theory, Coding and Cryptography”, vol.29, No.3, 2004, pp. 241-256

[3]. Lisa M. Marvel, Charles G. Boncelet, ‘Spread Spectrum Image Steganography’, in IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 8, NO. 8, AUGUST 1999, pp. 1075-1083

[4]. H.-W. Tseng, C.-P. Hsieh, Reversible data hiding based on image histogram modification, Imaging Science Journal vol. 56, Issue 5, 2008, pp. 271–278.

[5]. C.-C. Chang, T.-C. Lu, Y.-F. Chang, C.-T. Lee, Reversible data hiding schemes for deoxyribonucleic acid (DNA) medium, Intl. Journal Innovative Computing Information and Control 3 (5), 2007, pp. 1145–1160.

[6]. Z. Ni, Y.-Q. Shi, N. Ansari, W. Su, “Reversible data hiding”, IEEE Trans. on Circuits and Systems for Video Technology vol. 16, Issue 3, 2006, pp. 354–362.

[7]. Li YC, Yeh CM, Chang CC. “Data hiding based on the similarity between neighboring pixels with reversibility”, Digital Signal Processing (20), 2009, pp. 1116–1128

[8]. Z. Zhao et al.” Reversible data hiding based on multilevel histogram modification and sequential recovery”, electronics journal of Electronic communication (AEU), vol. 65, 2011, pp. 814-826

(a) (b)

Figure: Lena – Histogram