# Evaluating the Performance of the Secure Block Permutation Image Steganography Algorithm

**Article** *in* International Journal of Network Security & Its Applications · September 2013

DOI: 10.5121/ijnsa.2013.5514

**3 authors**, including:

Adnan Mahmood Shihab
University of Baghdad
**3** PUBLICATIONS **2** CITATIONS
SEE PROFILE

Raghad Khalied
University of Baghdad
**12** PUBLICATIONS **9** CITATIONS
SEE PROFILE

# EVALUATING THE PERFORMANCE OF THE SECURE BLOCK PERMUTATION IMAGE STEGANOGRAPHY ALGORITHM

Adnan M. Shihab, Raghad K. Mohammed, and Woud M. Abed

University of Baghdad, Baghdad, Iraq

## *ABSTRACT*

*Recently, a new secure steganography algorithm has been proposed, namely, the secure Block Permutation Image Steganography (BPIS) algorithm. The new algorithm consists of five main steps, these are: convert the secret message to a binary sequence, divide the binary sequence into blocks, permute each block using a key-based randomly generated permutation, concatenate the permuted blocks forming a permuted binary sequence, and then utilize a plane-based Least-Significant-Bit (LSB) approach to embed the permuted binary sequence into BMP image file format. The performance of algorithm was given a preliminary evaluation through estimating the PSNR (Peak Signal-to-Noise Ratio) of the stego image for limited number of experiments comprised hiding text files of various sizes into BMP images. This paper presents a deeper algorithm performance evaluation; in particular, it evaluates the effects of length of permutation and occupation ratio on stego image quality and steganography processing time. Furthermore, it evaluates the algorithm performance for concealing different types of secret media, such as MS office file formats, image files, PDF files, executable files, and compressed files.*

## *KEYWORDS*

*Steganography, permutation, encryption, steganalysis, LSB steganography, BMP image file*

## 1. INTRODUCTION

Steganography is the field of science that is concerned with hiding of information inside any media file in ways that prevent the disclosure of the hidden information to unauthorized recipients [1]. It is widely used as an information security approach to secure stored data or data exchanged over non-secured communication channels [2]. Steganography conveys the information secretly by concealing the very existence of information in some other media files such as image, audio, video, text files, or any other files. The information to be concealed is called the secret message or simply the secret; the media used to embed the secret is called the cover media, and the cover along with the secret are called the stego media [3].

Steganography has received a significant attention from many researchers throughout the world, especially, after the tremendous development in computer and Internet technologies, and the growing concern about information security. Subsequently, many steganography approaches have been proposed and used to develop a huge number of steganography algorithms. In particular, there are four basic broad approaches that can be used to accomplish steganography; these are: Lease-Significant-Bit (LSB), injection, substitution and generation approaches [4, 5].

Steganalysis is the art of identifying steganography by inspecting various parameters of stego media. After steganalysis determines the existence of hidden message, a steganography attacks

DOI : 10.5121/ijnsa.2013.5514

may be initiated to extract the secret message from the stego media or destroy it. As a result of that more secure steganography techniques are required [6-8]. One possible approach is combining cryptography and steganography, where cryptography can be used to conceal the contents of the secret, and steganography conceals the existence of the secret [9].

Many information security algorithms have been developed combining encryption and steganography algorithms to enhance information security [10]. One of the most recent algorithms is the secure Block Permutation Image Steganography (BPIS) algorithm [11]. The algorithm comprises five main steps; these are: convert the secret to a binary sequence, divide the binary sequence into blocks of length *N*, permute each block using a key-based randomly generated permutation *P* of length *N*, concatenate the permuted blocks to form a permuted binary sequence, and, finally, embed the permuted binary sequence into a cover image using the LSB approach. In [11], the algorithm performance is evaluated considering a limited number of experiments covering hiding text files of various sizes into BMP images.

This paper presents a wider investigation and performance evaluation; in particular, it evaluates the effects of length of permutation (*N*) and occupation ratio (*R*) on stego image quality and steganography processing time. Furthermore, it evaluates the algorithm performance for hiding different types of secret files, such as MS office files (*.docx, *.pptx, *.xlsx, image files (*.bmp, *.png, and *.jpg), PDF files (*.pdf), executable files (*.exe), and compressed files (*.zip). The PSNR between the stego and cover images was computed to estimate the distortion in stego image quality. The experimental results demonstrate that increasing *N* has insignificant effects on the stego image quality on one hand, and on the other hand almost linearly increases stenography processing time, which makes it hard to attack the BPIS algorithm.

This paper is divided into six sections. This section introduces the main theme of the paper. In Section 2, a literature review and summary of some of the most recent and related work are provided. A description of the BPIS algorithm is given in Section 3. The performance measures that are used in evaluating the performance of the BPIS algorithm are given in Section 4. A number of image steganography experiments are presented and discussed in Section 5. Finally, conclusions and recommendations for future research are pointed-out in Section 6.

## 2. LITERATURE REVIEW

This section provides an overview on the main components of a typical steganography system and briefly introduces current steganography approaches. Then, it reviews some of the most recent algorithms that have developed using these approaches, for the purpose of identifying features, advantages, and disadvantages of these algorithms.

### 2.1. Overview

Steganography is the art of hiding information by inserting a hidden (secret) message within other media files (e.g., images, audio, video or any other media) [1]. A typical steganography system consists of three main components, namely, secret, cover media, and stego media [12]. For a secure steganography, a forth components is required, which is the key or the password. The basic components of a secure steganography system are shown in Figure 1.

There are four basic steganography approaches that have been identified and widely used to accomplish steganography; these are [4, 5]:

1. *Lease Significant Bit (LSB) approach*, in which the LSBs of each byte of the cover media are replaced with bits from the message. The LSBs are also known as the stego bits.

2. *Injection approach*, in which the source message is hidden within the cover media that are ignored by the processing application. Therefore, avoid modifying those media bits that are relevant to an end-user leaving the cover media perfectly usable.

3. *Substitution approach*, in which the least significant meaningful content of the cover media is replaced with the source message in a way that causes the least amount of distortion to the cover media.

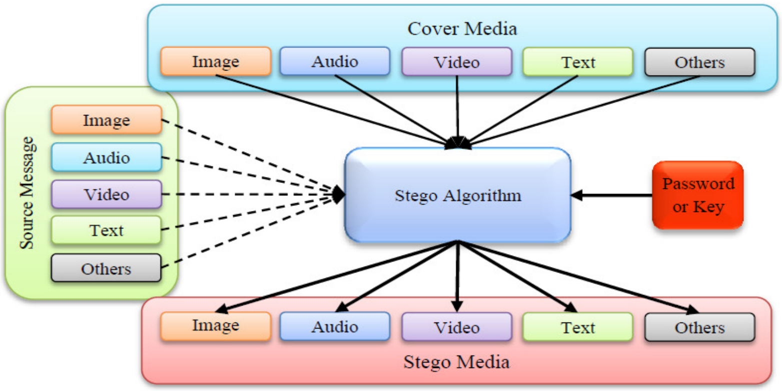

Figure 1. Basic components of secure steganography system [11].

4. *Generation approach*, which is different from the injection and substitution approaches; it does not require an existing cover media but generates a cover media for the sole purpose of hiding the message.

The above approaches have been widely used by a large number of researchers to develop various steganography algorithms. In this paper, we concern with the first steganography approach, the LSB approach for developing secure block permutation image steganography algorithm; therefore, next we shall review some of the most recent work on image steganography algorithms utilizing the LSB approach.

## 2.2. Previous Work

A LSB image steganography that enhances the existing LSB substitution algorithms was introduced in [10] to improve the security level of secret message. It encrypts the secret message to protect it from being accessed by unauthorized users before being hidden within the LSB of the image. The PSNR of the stego image was estimated to measure the stego images quality. The obtained results demonstrated that using secret key cryptography provides good security and PSNR value higher than general LSB based image steganography methods. Similarly, two algorithms combining cryptography and steganography were introduced in [9], in which the secret message is encrypted before being hidden. Although, such algorithms can provide higher resistance to steganalysis, they usually take long processing time.

In [13] a steganography algorithm using non-uniform adaptive image segmentation with an intelligent computing technique was presented to conceal large secret messages into color images. The algorithm can provide a high capacity of up to 4-bit per byte while maintaining image visual quality. A steganography algorithm that divides the cover image into blocks and embeds the corresponding secret data bits into each block without any permutation was developed in [14]. In this algorithm, for each secret bit sequence, it performs a search on the rows and columns of the layers to find the most similar row or column. Thus, the algorithm introduces a very low image distortion in comparison to other existing algorithms.

A novel Genetic Algorithm (GA) evolutionary process was developed in [15] to secure steganography encoding on JPEG images. A steganography algorithm for hiding text message using the LSB approach along with the concept of chaos theory was described in [12]. The algorithm provides security and maintains secrecy of the secret and provides more randomness. A general LSB substitution model called the transforming LSB substitution model was developed in [20] to embed secret data in LSBs of pixels in a cover image. An image steganography algorithm for hiding a message into a gray level image was developed in [17]. In order to maximize the storage of data inside the image, a steganography algorithm was proposed in [18] that utilized a pre-compression step. Further literature review can be found in [11].

## 3. The Secure BPIS Algorithm

This section presents description of the proposed BPIS algorithm [11]. The algorithm makes use of BMP image file as a cover media (cover image) to hide a secret message. It utilizes the LSB approach, which is a simple way of steganography, where it replaces one or more of the LSBs of the image pixels with bits from the secret message. The LSB approach either directly embeds the secret message inside the planes of the pixels of the cover image, or, in some cases, the LSBs of the pixels are visited in random or in certain positions of the image [10].

In order to be increase the immunity of the algorithm against steganalysis, the BPIS algorithm is designed to have two main procedures, these are:

1. Security or pre-steganography procedure.
2. Steganography procedure.

Figure 2 shows the main components of the BPIS algorithm and a brief description of these two procedures are given next.

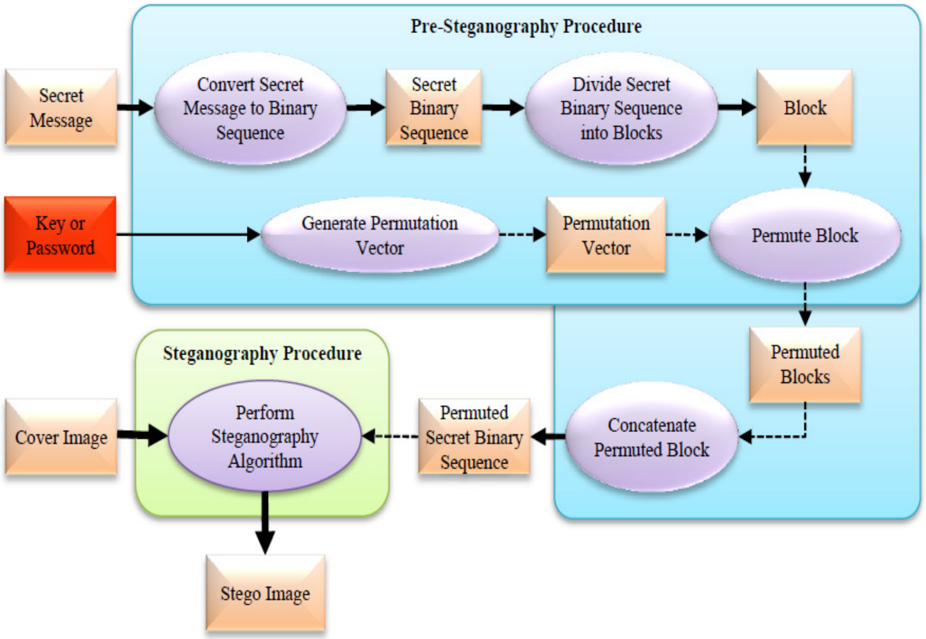
Pre-Steganography Procedure
Secret Message
Convert Secret Message to Binary Sequence
Secret Binary Sequence
Divide Secret Binary Sequence into Blocks
Block
Key or Password
Generate Permutation Vector
Permutation Vector
Permute Block
Permuted Blocks
Steganography Procedure
Cover Image
Perform Steganography Algorithm
Permuted Secret Binary Sequence
Concatenate Permuted Block
Stego Image

Figure 2. The main components of the BPIS algorithm [11].

## 3.1. Security Procedure

A security procedure is proposed to secure the secret message before proceeding with the hiding it inside the cover image. The procedure, first, converts the secret message to a binary sequence (secret binary sequence) using a certain character-to-binary conversion technique (such as ASCII codes), where each character is converted to 8-bit binary representation equivalent to its ASCII code and concatenated with binary representation of previous characters until all characters of the secret message is proceed. Second, it divides the secret binary sequence into fixed-size blocks of length *N*. Third, It permute each block using key-based randomly generated permutation, and if the number of bits in the last block is less than *N*, then leave the block as it is. Finally, in the fourth step, it reconstructs the secret binary sequence by concatenating the permuted blocks to form a permuted binary sequence, which represent the permuted secret binary sequence.

The algorithm utilizes a simple password-based or key-based procedure to generate a permutation *P* of length *N*. The permutation *P* is generated randomly using any Random Number Generator (RNG). The same RNG and key or the password (*K*) must be used to determine *P* during the secret message recovery process. The security of the BPIS algorithm depends on *N*, and the time complexity is $O(N!)$. In other word, performing a brute-force attack to determine *P* requires *N!* trials. Although, a user can insert any appropriate integer value for *N*, it is preferable to select *N* such that $N \geq 256$.

## 3.2. Steganography Procedure

The steganography approach used in the BPIS algorithm is the LSB, which modifies the LSBs of the pixels of the cover image (cover Bytes) [10, 12, 16]. In LSB approach, the bits of the permuted secret binary sequence are distributed among the LSBs of consecutive pixels, starting from the Red plane (R-plane) of the top-left pixel moving towards the right modifying consecutive pixels on the row, and then move to the next row on the same plane and so on until modifying the 1st bit of the R-plane for all pixels. If the R-plane is not enough to host all secret bits, then the process continues in the same way on the Green and Blue planes (G-plane and B-plane). If it is still not enough, repeat the procedure above replacing the 2nd LSBs of the RGB planes of all pixels, and continue replacing the 3rd LSBs if the 2nd LSBs of the RGB planes are enough to host the permuted sequence. The process terminated when all secret bits are hidden inside the cover Bytes of the cover image, subject to the constraint that the number of modified LSBs should be not more than 3.

The maximum image capacity ($C_{max}$) in Byte of the BPIS algorithm can be determined as [11]:

$$C_{max} = (s \cdot p \cdot W \cdot H) / b \tag{1}$$

Where, $s$ represents the number of stego bits (limited to 3-bit in the BPIS algorithm), $p$ represents the number of color planes (3 for RGB color image), $W$ and $H$ represent the width and height of the cover image in pixels, $b$ represents the number of bits per Byte (bpb). For $s$=3, $p$=3, $b$=8, $C_{max}$ can be calculated as:

$$C_{max} = 1.125\ W \cdot H \tag{2}$$

Another parameter was introduced in [11], namely, the occupation time ($R$), which is calculated by dividing the sizes of the hidden message ($S$) (secret message plus the stego header) by the maximum allowable image capacity ($C_{max}$). It is expressed mathematically as:

$$R = \frac{S}{C_{max}} \text{x } 100 \tag{3}$$

Where $R$ is the occupation ratio, and $S$ is the actual size of the hidden message ($S$=M+$D$, where $M$ is the size of the secret message and $D$ is the size of the stego header).

## 4. PERFORMANCE MEASURES

The performance of the BPIS algorithm is evaluated by estimating the PSNR, which measures the distortion in the stego image quality. In fact, the PSNR measures the variation between the stego and cover images and it is expressed in terms of the logarithmic decibel (dB) scale as [12]:

$$PSNR = 20 \cdot log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \tag{4}$$

Where, $MAX_I$ is set to 255, and MSE (Mean Square Error) is the cumulative squared error between the stego and cover images, and for 24-bit BMP images it can be calculated as [12, 18]:

$$MSE = \frac{1}{W \cdot H \cdot P} \sum_{p=1}^{P} \sum_{h=1}^{H} \sum_{w=1}^{W} [S(w,h,p) - C(w,h,p)]^2 \tag{5}$$

Where $S(w,h,p)$ and $C(w,h,p)$ are the Byte values at plane $p$ and position $w$,$h$ on the stego and cover images, $P$ is the number of planes, and $W$ and $H$ are the width and height of the images. It is clear from Eqn. (4) that the larger the PSNR the smaller is the difference between stego and cover images, which means maintaining the image quality and at the same time makes it difficult for the attacker to discover the existence of the secret message inside the stego image. For an algorithm using three stego bits and RGB color image, the maximum MSE value is 7, which yields a minimum PSNR of 31.23 dB.

## 5. Results and Discussions

The main objective of this paper is to evaluate the performance of the BPIS algorithm. In this direction, we conducted two sets of experiments as summarized below.

### 5.1. Experimental Results for Set #1

This set comprises a number of experiments to evaluate the performance of the algorithm as follows:

1. Investigate the effect of *N* on the stego image quality by estimating the PSNR between the stego and cover images for standard text files of various sizes.
2. Estimate the following CPU times:
   a. Total processing time ($T_{tot}$), which represents the CPU time required to accomplish hiding the secret message inside the cover image.
   b. Permutation generation time ($T_{gen}$), which represents the CPU time required to compute a permutation *P* of length *N*.
   c. Block permutation time ($T_{per}$), which represents the CPU time required to permute the secret bits.
   d. Steganography time ($T_{stego}$), which represents the CPU time required to conceal the secret message inside the cover image. It also includes CPU time for reading the cover image, converting cover Bytes to binary sequence, covert stego sequence to stego Bytes, and save the stego Bytes into the stego image.

Thus, $T_{tot}$ can be expressed as follows:

$$T_{tot} = T_{gen} + T_{per} + T_{stego} \tag{6}$$

In each experiment a standard text file is used as a secret and concealed inside the cover image using the BPIS algorithm with different *N*. In particular, five permutation sizes *N* are considered; these are, 0 (no permutation), 64, 128, 256, and 512. The cover image used is the Flowers.BMP. It is three planes image with image width *W*=500 pixels and image height *H*=362 pixels. The total size of the image is 543054 Bytes, 54 Byte for the image header and the remaining 543000 Byte are pixel colors. In this case, the maximum image capacity $C_{max}$ is 203625 Byte. The results of the PSNR are given in Table 1, which also shows the sizes of the text files and the occupation ratio (*R*). The results for PSNR are also plotted in Figure 3.

Table 1. Variation of PSNR with *N* for various text files as secret messages.

| **Filename** | **Size (Byte)** | ***R* (%)** | **Length of Permutation (*N*)** | | | | |
|---|---|---|---|---|---|---|---|
| | | | **0** | **64** | **128** | **256** | **512** |
| Paper1 | 53161 | 26.1 | 52.211 | 52.080 | 52.220 | 52.227 | 52.207 |
| Paper2 | 82199 | 40.4 | 49.032 | 49.037 | 49.038 | 49.029 | 49.029 |
| Asyoulik | 125179 | 61.5 | 45.675 | 45.670 | 45.666 | 45.675 | 45.667 |
| Alice29 | 152089 | 74.7 | 42.787 | 42.797 | 42.777 | 42.784 | 42.785 |
| RFCs | 195272 | 95.9 | 39.376 | 39.379 | 39.367 | 39.371 | 39.368 |

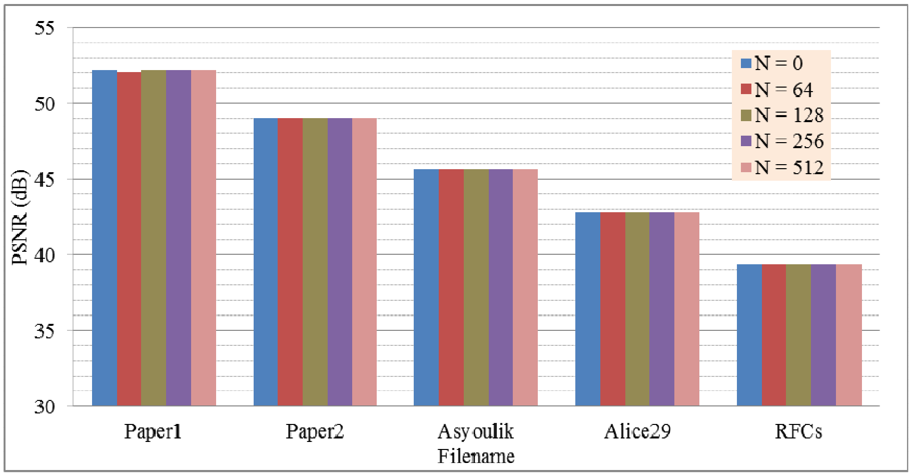

Figure 3. Variation of PSNR with *N* for various text files as a secret messages.

For each of the above experiments, we also computed the CPU times $T_{tot}$, $T_{gen}$, $T_{per}$, and $T_{stego}$ discussed above. Table 2 lists the results for $T_{per}$ and $T_{tot}$. $T_{gen}$ depends on *N*, practically, it is directly proportional to *N*, and it is computed as 0.016, 0.031, 0.047, and 0.11 sec for *N*=64, 128, 256, and 512 for all experiments. Finally, the experimental results show that $T_{stego}$ is a function of the size of the secret and independent of *N*. In particular, it is equal to $T_{tot}$ when *N*=0, which is expected as in this case $T_{gen}$ and $T_{per}$ are equal to 0, and if substituted in Eqn. (6), then $T_{tot}$=$T_{stwgo}$. The timing results are for Intel® 2.2 GHz Core i7 CPU with installed memory of 6 GB, and running Windows 7 Professional 64-bit operating system.

Table 2. The CPU times of $T_{per}$ and $T_{tot}$ for experiments of Set #1.

| **Filename** | **$T_{per}$ (sec)** | | | | | **$T_{tot}$ (sec)** | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **0** | **64** | **128** | **256** | **512** | **0 ($T_{stego}$)** | **64** | **128** | **256** | **512** |
| Paper1 | 0.00 | 38.42 | 38.17 | 38.10 | 38.43 | 12.00 | 51.85 | 49.48 | 49.26 | 49.97 |
| Paper2 | 0.00 | 113.26 | 112.51 | 112.51 | 112.67 | 32.92 | 149.44 | 145.28 | 145.37 | 145.51 |
| Asyoulik | 0.00 | 280.57 | 280.94 | 280.38 | 280.22 | 77.46 | 358.01 | 358.33 | 357.86 | 357.78 |
| Alice29 | 0.00 | 431.07 | 428.53 | 428.35 | 429.16 | 118.90 | 550.37 | 547.40 | 546.95 | 547.58 |
| RFCs | 0.00 | 749.35 | 747.41 | 747.97 | 747.07 | 199.82 | 999.75 | 948.40 | 949.38 | 950.41 |

The outcomes and discussions of the above experiments can be summarized as follows:

1. The permutation has insignificant effect on the stego image quality as the PSNR remains unchanged regardless of *N*.
2. The stego image quality decreases as more stego bit are required to accommodate the secret sequence (i.e., increasing *R*), which is very much expected as the variation may increase with increasing number of stego bits. However, in all cases the stego image still maintain acceptable quality and at the same time visual inspection shows excellent invisibility.
3. For the same secret (text) file, $T_{per}$ and $T_{tot}$ remain unchanged with increasing *N*, because the size of the secret message is unchanged. However, they increase with increasing secret size. It can be easily realized that $T_{tot}$ for permuted secret steganography is almost 4-5 times the non-permuted secret steganography. This makes attacking the BPIS algorithm harder.

## 5.2. Experimental Results for Set #2

This set of experiments determines the PSNR between the stego and covers images for concealing different types of secret media files inside the same cover image described in Section 5.1. In particular, we conducted a number of experiments for concealing a list of various media files. The description, size of each media file, occupation ratio computed with reference to a maximum of 3 LSBs, and the PSNR of the stego are given in Table 3.

Table 3. Results for experimental Set #2.

| # | Media | Description | Size (Byte) | R (%) | PNSR (dB) |
|---|---|---|---|---|---|
| 1 | *.docx | MS Word Document | 189644 | 93.1 | 39.840 |
| 2 | *.xlsx | MS Excel Worksheet | 123895 | 60.9 | 45.742 |
| 3 | *.pptx | MS Power Point Presentation | 189860 | 93.2 | 39.817 |
| 4 | *.pdf | Adobe Acrobat Document | 191238 | 93.9 | 39.731 |
| 5 | *.exe | Application | 188872 | 92.8 | 39.879 |
| 6 | *.bmp | BMP image | 198774 | 97.6 | 39.341 |
| 7 | *.jpg | JPEG image | 188577 | 92.6 | 39.892 |
| 8 | *.png | PNG image | 189240 | 92.9 | 39.852 |
| 9 | *.zip | WinRAR ZIP archive | 160744 | 78.9 | 41.890 |

The results presented in Table 3 show that the stego image maintain the same image quality regardless of the type of the secret media, and it only depends on the size of the secret media or occupation ratio. Since, we demonstrated in the previous set of experiments that *N* has insignificant effect on the PSNR, *N* is set to 128 for this set of experiments. These results are very encouraging to use the PSNR as a powerful and secure steganography algorithm.

## 6. CONCLUSIONS

This paper presented a comprehensive performance evaluation for the new secure BPIS algorithm. In particular, it presented two sets of experiments to evaluate the performance of the algorithm. The main outcomes of these two sets of experiments are: (1) The permutation has insignificant effect on the stego image quality as the PSNR remains unchanged regardless of *N*. (2) The stego image quality decreases with increasing occupation ratio as more stego bit are required to accommodate the secret sequence. (3) Increasing *N* has no effect on the permutation and total CPU times. (4) The total CPU time for the BPIS algorithm is about 4 to 5 times the non-permuted algorithms (*N*=0) regardless of the size of the secret message. (5) The type of the secret media has insignificant effect on the estimated PSNR and it is only affected by the size of the media file. (6) In all experiments, the stego image maintain acceptable image quality and at the same time visual inspections show excellent invisibility. (7) The relatively high processing time of the BPIS algorithm makes it hard to be attacked.

## REFERENCES

[1] Zoran Duric, Michael Jacobs, and Sushil Jajodia. Information Hiding: Steganography and Steganalysis. Review Article Handbook of Statistics, Vol. 24, pp. 171-187, 2005.

[2] Atallah M. Al-Shatnawi. A New Method in Image Steganography with Improved Image Quality. Journal of Applied Mathematical Sciences, Vol. 6, No. 79, pp. 3907-3915, 2012.

[3] Rajkumar Yadav, Ravi Saini, and Kamaldeep. Cyclic Combination Method for Digital Image Steganography with Uniform Distribution of Message. An International Journal on Advanced Computing (ACIJ), Vol. 2, No. 6, pp. 29-43, November 2011.

[4] T. Morkel, J. H. P. Eloff, and M. S. Olivier. An Overview of Image Steganography. Proceedings of the 5th Annual Information Security South Africa Conference (ISSA2005) (Eds.: H. S. Venter, J. H. P. Eloff, L. Labuschagne, and M. M. Eloff), Sandton, South Africa, 2005. Retrieved from http://martinolivier.com/open/stegoverview.pdf.

[5] Nagham Hamid, Abid Yahya, R. Badlishah Ahmad, and Osamah M. Al-Qershi. Image Steganography Techniques: An Overview. International Journal of Computer Science and Security (IJCSS), Vol. 6, Issue 3, pp. 168-187, 2012.

[6] V. Natarajan and R Anitha. Blind Image Steganalysis Based on Contourlet Transform. International Journal on Cryptography and Information Security (IJCIS), Vol. 2, No. 3, pp. 77-87, September 2012.

[7] G. R. Xuan, Y. Q. Shi, J. J. Gao, D. Zou, C. Yang, Z. Zhang, P. Chai, C. Chen, and W. Chenl. Steganalysis based on Multiple Features Formed by Statistical Moments of Wavelet Characteristic Functions. Proceedings of the 7th International Information Hiding Workshop, LNCS, Vol. 3727, pp. 262-277, Springer-Verlag, 2005.

[8] W. N. Lie and G. S. Lin. A Feature-based Classification Technique for Blind Image Steganalysis. IEEE Transaction on Multimedia, Vol. 7, No. 6, pp. 1007-1020, 2005.

[9] Sujay Narayana and Gaurav Prasad. Two New Approaches for Secured Image Steganography Using Cryptographic Techniques and Type Conversions. The International Journal of Signal & Image Processing (SIPIJ), Vol.1, No.2, pp. 60-73, December 2010.

[10] S. M. Masud Karim M. S. Rahman, and M. I. Hossain. A New Approach for LSB Based Image Steganography using Secret Key. Proceedings of 14th International Conference on Computer and Information Technology, IEEE Conference Publications, pp. 286 – 291, 2011.

[11] Hussein Al-Bahadili. A Secure Block Permutation Image Teganography Algorithm. Submitted to the International Journal on Cryptography and Information Security (IJCIS) on 30th July 2013.

[12] S. Bhavana and K. L. Sudha. Text Steganography Using LSB Insertion Method Along with Chaos Theory. International Journal of Computer Science, Engineering and Applications (IJCSEA), Vol.2, No.2, pp. 145-149, April 2012.

[13] Nameer N. El-Emam, Rasheed Abdul Shaheed Al-Zubidy. New Steganography Algorithm to Conceal a Large Amount of Secret Message Using Hybrid Adaptive Neural Networks with Modified Adaptive Genetic Algorithm. Journal of Systems and Software, Vol. 86, Issue 6, pp. 1465-1481, June 2013.

[14] O. Kurtuldu and N Arica. A New Steganography Method Using Image Layers. Proceedings of the 23rd International Symposium on Computer and Information Sciences (ISCIS '08), pp. 1-4, Istanbul, Turkey, 27-29 October 2008.

[15] A. M. Fard, M. M. R. Akbarzadeh-T, and F. Varasteh-A. A New Genetic Algorithm Approach for Secure JPEG Steganography. Proceedings of the IEEE International Conference on Engineering of Intelligent Systems, pp. 1-6, Islamabad, Pakistan, 2006.

[16] Cheng-Hsing Yang and Shiuh-Jeng Wang. Transforming LSB Substitution for Image-based Steganography in Matching Algorithms. Journal of Information Science and Engineering (JISE), Vol. 26, pp. 1199-1212, 2010.

[17] Rajkumar Yadav, Ravi Saini and Kamaldeep. Cyclic Combination Method for Digital Image Steganography with Uniform Distribution of Message. International Journal of Advanced Computing (ACIJ), Vol. 2, No. 6, pp. 29-43, November 2011.

[18] Rosziati Ibrahim and Teoh Suk Kuan. Steganography Algorithm to Hide Secret Message inside an Image. Journal of Computer Technology and Application, Vol. 2, pp. 102-108, 2011.

## Authors

Adnan M. Shihab is working as a lecturer at the Department of Basic Sciences, College of Dentistry, University of Baghdad (Baghdad, Iraq). Mr. Shihab is also acting as the director of Human Resources at the College of Dentistry. He received his B.Sc degrees in Computers from College of Education, University of Al-Mustansiriya, Baghdad, Iraq in 1993, and his M.Sc degree in Computers from the Department of Computers and Informatics, University of Technology, Baghdad, Iraq in 2007. His research interest covers block cipher, steganography, authentication, computer networks, and image processing.

Raghad K. Mohammed is serving as a member of academic staff at the Department of Basic Sciences, College of Dentistry, University of Baghdad (Baghdad, Iraq). She received her B.Sc degree in Computer Science from the Department of Computer science, Alrafidain University College (Baghdad, Iraq) in 2003, and her M.Sc degree in Computer Networks, Informatics Institute for Higher Studies, University of Technology (Baghdad, Iraq) in 2005. Her research interests include cryptography and steganography, image processing, and information and network security.

Woud M. Abed is a member of academic staff at the Department of Basic Sciences, College of Dentistry, University of Baghdad (Baghdad, Iraq). She received her B.Sc degree in Computer Science from the Department of Computer Science, Alrafidain University College (Baghdad, Iraq) in 2003, and her M.Sc degree in Computer Networks, Informatics Institute for Higher Studies, University of Technology (Baghdad, Iraq) in 2005. Her research interests include: robotic, genetic algorithms, cryptography and steganography, image processing, and computer security.