A SECURE BLOCK PERMUTATION IMAGE STEGANOGRAPHY ALGORITHM

Hussein Al-Bahadili

Faculty of Information Technology, University of Petra, Amman, Jordan

ABSTRACT

Steganography is the art of hiding confidential information (secret) within any media file (cover media) to produce an amalgamated secret-cover media called stego media, so that the secret cannot be recognized or recovered by unauthorized recipients. Many steganalysis techniques have been developed enabling recognition of the existence of secrets within stego media and recovering it. Therefore, it is necessary to develop more secure steganography algorithms. This paper presents a detailed description of a new secure Block Permutation Image Steganography (BPIS) algorithm. The algorithm converts the secret message to a binary sequence, divides the binary sequence into blocks, permutes the block using a key-based randomly generated permutation, concatenates the permuted blocks forming a permuted binary sequence, and then utilizes the Least-Significant-Bit (LSB) approach to embed the permuted binary sequence into BMP image file. The algorithm performance is investigated through performing a number of experiments, and for each experiment the PSNR (Peak Signal-to-Noise Ratio) between the stego and cover images is calculated. The results show that the algorithm provides high image quality, and invisibility, and most importantly higher security as secret cannot be recovered without knowing the permutation, which has a complexity of O(N!), where N is the length of the permutation.

KEYWORDS

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

1. INTRODUCTION

Steganography is defined as the art of hiding information within any media file in ways that prevent the disclosure of the hidden information to unauthorized recipients [1, 2]. Thus, it can be used as an information security approach to secure stored data or data exchanged over non-secured communication channels [3]. Steganography conveys the information secretly by concealing the very existence of information in some other media files such as image, audio, video, or text files. The information to be concealed is called the secret message or simply the secret; the content used to embed information is called the cover media, and the cover along with the secret is called the stego media [4].

Due the tremendous development in computer and Internet technologies, and the growing concern about information security, steganography has received significant attention from both academia and industry. Consequently, a number of 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 [5-9].

At the same time, another field of interest, namely, steganalysis is introduced to identify steganography by inspecting various parameters of stego media. After steganalysis determines whether a media contains hidden message or not, a steganography attacks 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 [10-12].

Encryption is a well-known procedure for securing data transmission or storage. Many encryption methods have been developed throughout the years, such as: DES (Data Encryption Standard), AES (Advanced Encryption Standard) and RSA [14]. These methods scramble the secret message so that it cannot be understood. However, it makes the message suspicious enough to attract eavesdropper's attention. Hence, steganography looks as a good alternative approach as it conceals the secrets within some other ordinary media files so that it cannot be observed. In summary, steganography differs from cryptography in the sense that where cryptography conceals the contents of a message, steganography conceals the existence of a message [3].
Many information security algorithms have been developed combining both encryption and steganography algorithms to enhance information security [7]. So that if an attacker succeeds in detecting and extracting the secret, he/she will find it encrypted. If the encryption algorithm is known, then using brute-force attack to decrypt the secret has a complexity of $O(2^N)$, where N is the length of the encryption key. In this work, we develop a new secure steganography algorithm that utilizes the concept of permutation. Permutation is defined as the act of changing the arrangement of a given number of elements, and it is widely-used as part of many encryption algorithms, such as DES, AES, RSA, etc [14]. Permutation can be performed faster than encryption and at the same time it has higher resistivity to brute-force attack, where the complexity of permutation of length N is $O(N!)$ because there are N! possible permutations of a string [14].

This paper presents detail description of a new secure Block Permutation Image Steganography (BPIS) algorithm. 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 file using a LSB approach. The algorithm performance is investigated through experiments. In particular, the algorithm is used to hide text files of different sizes into BMP images of various sizes, and the PSNR between the stego and cover images is calculated. The experimental results show that the algorithm maintains image quality and good invisibility as it provides PSNRs of more than 40 dB. Furthermore, satisfactory security is maintained since the secret cannot be recovered without knowing the permutation.

This paper is developed into five sections. Section 1 introduces the main theme and concept of the paper. Section 2 provides a literature review and reviews some of the most recent and related work. A detail description of the BPIS algorithm is given in Section 3. The results of using the algorithm in a number of image steganography applications are presented and discussed in Section 4. Finally, in the Section 5, conclusions are drawn and a number of recommendations for future research are pointed-out.

2. LITERATURE REVIEW

This section present a review on some of the most recent and related work. But, before we proceed, let us describe the basic components of steganography system and basic steganography approaches. A steganography system usually consists of three main components, namely, secret, cover media, and stego media [19]. For a secure steganography system, a forth components is required, which is the key or the password. In general, the secret and the cover media can have

the form of image, audio, video, text file or other media file. Figure 1 presents a schematic showing the basic components of a secure steganography system. For the steganography approaches, there are four basic broad approaches that can be used to accomplish steganography; these are [5-9]:

(1) *Lease Significant Bit (LSB) approach.* In this approach, the LSB of each byte of the cover file is replaced with bits from the message.
(2) *Injection approach.* In this approach, the source message is hidden in sections of the cover file that are ignored by the processing application. Therefore, avoid modifying those file bits that are relevant to an end-user leaving the cover file perfectly usable.
(3) *Substitution approach.* In this approach, the least significant meaningful content of the cover file is replaced with the source message in a way that causes the least amount of distortion to the cover file.
(4) *Generation approach.* It is unlike injection and substitution; it does not require an existing cover file but generates a cover file for the sole purpose of hiding the message.

A large number of steganography algorithms have been developed utilizing the above approaches. In this paper, we concern with the first steganography approach, namely, the LSB approach for hiding text message in image file, therefore, in what follow we shall only review some of the most recent researches on image steganography.

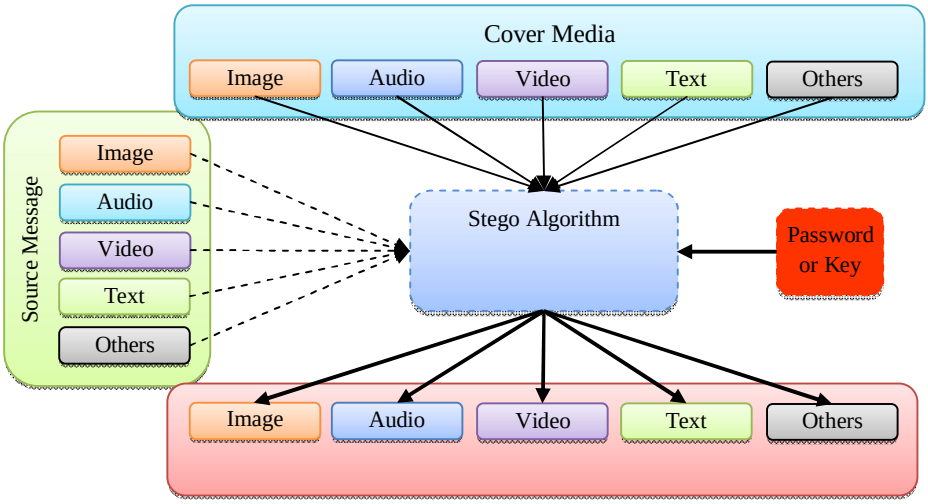


Figure 1. Basic components of secure steganography system.

Masud Karim et al. [7] introduced a LSB image steganography that enhances the existing LSB substitution techniques to improve the security level of hidden information. It encrypts the hidden information to protect it from unauthorized users before being hidden within the LSB of the image. They estimated the PSNR to measure the stego images quality. Results demonstrated that the proposed LSB based image steganography with secret key provides good security and PSNR value higher than general LSB based image steganography methods.

El-Emam et al. [15] presented a steganography algorithm using non-uniform adaptive image segmentation with an intelligent computing technique to conceal a large amount of confidential messages into color images. Their results demonstrate that the algorithm can efficiently hide a secret message with a capacity of up to four bits per byte while maintaining image visual quality. Kurtuldu and Arica [16] proposed a steganography method that divides the cover image into blocks and embeds the corresponding secret data bits into each block without any permutation.

Instead, for a given secret bit sequence, it performs a search on the rows and columns of the layers for finding the most similar row or column. Experimental results showed that the method introduces a very low image distortion in comparison to other existing steganography techniques.

Fard et al. [17] developed a novel Genetic Algorithm (GA) evolutionary process to secure steganography encoding on JPEG images. Narayana and Prasad [18] introduced two methods wherein cryptography and steganography are combined to encrypt the secret message as well as to hide the encrypted message in another medium. Although, such methods can provide higher resistance to steganalysis, but it take long processing time. Bhavana and Sudha [19] described a steganography algorithm for hiding text message using the LSB method along with the concept of non-linear dynamic system (chaos) theory. The algorithm provides security and maintains secrecy of the secret and provides more randomness. The performance of the algorithm is analyzed and the PSNR value is also being calculated to indicate the effect of the proposed algorithm on the image quality. Yang and Wang [20] developed a general LSB substitution model called the transforming LSB substitution model to embed secret data in LSBs of pixels in a cover image.

Yadav et al. [21] developed an image steganography algorithm for message hiding into a gray level image. The algorithm distributes the message uniformly throughout the image; where the image is divided into equal size blocks and the message is then hidden into the central pixel of the block using cyclic combination of 6^{th} to 8^{th} bit. The blocks of the image are chosen randomly using pseudorandom generator seeded with a secret key. The cyclic combination of last three bits of pixel value provide 100% chances of message insertion at the pixel value and division of image into blocks distribute the message uniformly into the image. This method also provides minimum degradation in image quality that cannot be perceived by human eye. Ibrahim and Kuan [22] proposed an image steganography technique that utilizes a pre-compression step to maximize the storage of data inside the image. The compressed file is then converted to binary codes and concealed within the image pixels. The algorithm is tested for hiding various sizes of data inside images and the PSNR is estimated for each of the images tested. They concluded that their steganography algorithm is very efficient in data hiding.

Farid [25] and Lyu and Farid [26] modeled a blind steganalyzer using supervised learning (SL) and indicated that the SL is effective for detecting stego images without knowing the statistical property of images and steganography methods. Xuan et al. [11] also presented a blind steganalysis method, which was based on statistical moments of wavelet histogram characteristic functions and Bayes classifier. Experimental results indicated that this method worked better for LSB methods. Lie et al. [12] indicated that in general no single feature is capable of differentiating stego and cover images effectively and a combination of features extracted in different domain will be generally more promising. By means of extracting features from spatial and DCT domain, this technique had a good effect for BMP images, including spatial domain and DCT domain hiding techniques. Luo et al. [27] proposed a blind steganalysis method with high detection ratio based on best wavelet packet decomposition. The methods based on wavelet high order statistics, so they cannot perform very well on LSB steganography.

Avcibas et al. [23] presented techniques for steganalysis of images that have been potentially subjected to a watermarking algorithm. Then, the same authors developed in [24] more universal techniques for steganalysis of images that have been potentially subjected to steganographic algorithms, both within the passive warden and active warden frameworks. Simulation results indicate that the new approach is able with reasonable accuracy to distinguish between cover and stego images. Sajedi et al. [13] presented a steganalysis called Contourlet Based Steganalysis (CBS), which uses statistical moments as well as the log errors between the actual and predicted coefficients of the contourlet transform as features for analysis. After feature extraction, a nonlinear SVM classifier is applied to classify stego and cover images. This method converts the

image into gray-scale and then processes it. CBS detection rate is very low when message is embedded in medium frequency sub-bands and this idea is used in [28] to develop a new contourlet based steganography algorithm.

3. THE SECURE BPIS ALGORITHM

This section presents description of the proposed BPIS algorithm. The algorithm makes use of BMP image file as a cover media (cover image) to hide the secret message; therefore, it is referred to as image steganography. The algorithm utilizes the LSB approach, which is a simple way of steganography. It is based on replacing one or more of the LSBs of the image pixels with bits from the secret message (usually not more than 4 bits). The LSB approach either directly embeds the secret message within 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 [7, 8].

As discussed in Section 2 that there are a number of steganalysis algorithms have been developed in an attempt to discover hidden data in suspected media or at least detect if a media contains hidden data. Therefore, it is necessary to think of securing the secret data before proceeding with the steganography process, and develop more secure steganography algorithms. The BPIS algorithm is one algorithm in this direction.
The BPIS algorithm consists of two main procedures, which are shown in Figure). These procedures are:

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

In what follows, we shall provide a description of these two procedures.

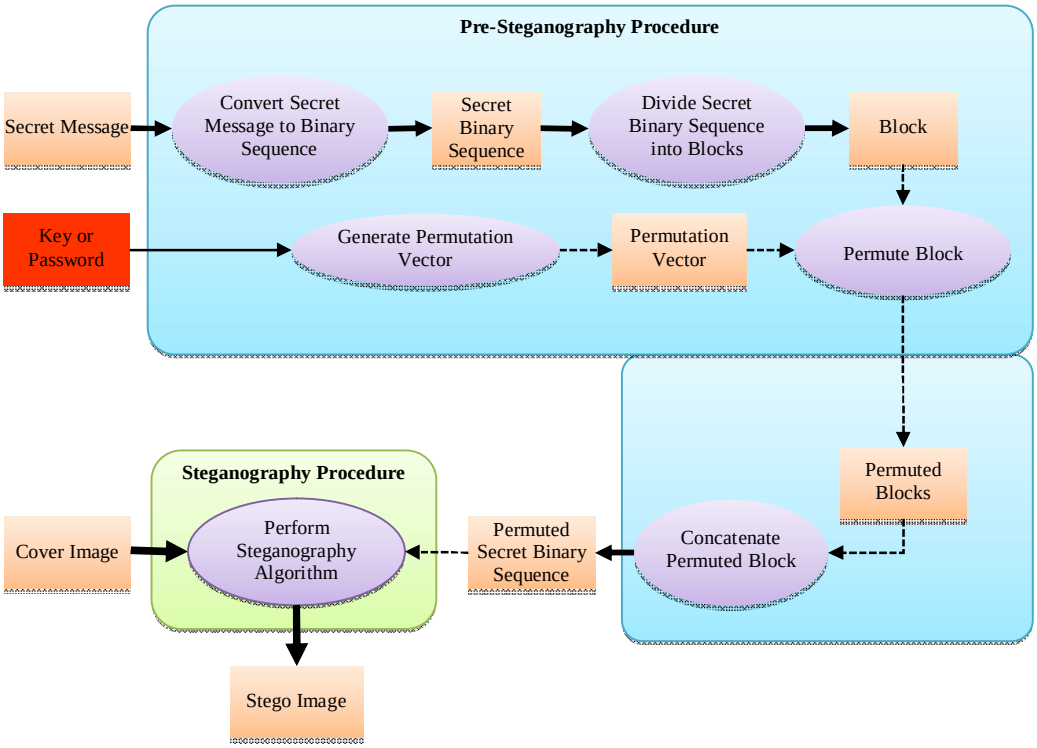


Figure 2. The main components of the BPIS algorithm.

3.1. Security or pre-steganography procedure

In this work, a pre-steganography procedure is proposed to secure the secret message before proceeding with the steganography procedure. The procedure comprises the following steps:

(1) Convert 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 converted.
(2) Divide the secret binary sequence into fixed-size blocks of length *N*.
(3) Permute each block using key-based randomly generated permutation. If the number of bits in the last block is less than *N*, then leave the block as it is.
(4) Concatenate the permuted blocks to form a permuted binary sequence, which represent the permuted secret binary sequence.

Then, in the next procedure, bits from the permuted secret binary sequence are hidden within LSBs of pixels from the cover image using the steganography procedure described below.

3.1.1. Generation of the permutation vector (*P*)

In this paper, a simple password-based or key-based permutation generation procedure is developed to generate a permutation *P* of length *N*. The permutation *P* is generated randomly using any Random Number Generator (RNG).

The elements of the permutation vector are determined using the following equation:

$$P_i = 1 + \lfloor 1 + \xi_1 \cdot N + \xi_2 \cdot N^2 + \xi_3 \cdot N^3 \rfloor \bmod N \qquad (1)$$

Where $\xi_{1,2,3}$ are random numbers ($0 \leq \xi < 1$), *N* is the size of the permutation, and P_is (*i*=1 to *N*) represent elements of permutation. After calculating P_1, a new element is calculated and it is accepted only if it is not predetermined, i.e., P_i is not equal to the value of any of the previous elements P_1 to P_{i-1}. The seed of the RNG is calculated based on a user inserted string of characters, denoted as the key or the password (*K*). The same RNG and *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 $N \geq 256$. Examples of different permutation of *N*=16 generated using the procedure described above for different *K*s are given in Table 1. It shows the sensitivity of the function in Eqn. (1) to any change in *K*.

Table 1. Examples of permutation vectors generated using the random procedure.

Password/Key (K)	Permutation															
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
COMPUTER	5	11	1	3	8	14	6	9	4	10	2	16	15	7	13	12
Computer	15	11	3	5	12	10	6	16	14	13	2	4	9	7	8	7
computer	10	4	5	15	12	8	2	11	16	7	3	6	14	9	13	1
comp2013	4	6	10	7	1	13	9	3	12	14	16	2	8	5	15	11
comp&013	14	8	4	6	1	3	7	11	5	13	16	15	2	12	10	9
computing	15	6	1	3	14	9	16	4	10	8	7	12	5	13	2	11
computers	8	1	13	3	9	7	5	14	15	11	6	10	2	12	16	4

3.2. Steganography procedure

The steganography approach used in this work is the LSB approach, which is based on modifying the LSBs of the pixels of the BMP cover image [3-4, 6-7]. In LSB approach, the bits of the permuted secret binary sequence are distributed among the LSBs of consecutive pixels, starting from the top-left pixel of the R-plane (Red plane) moving towards the right modifying consecutive pixels on the row, and then move to the next row and so on until modifying the 1st bit of all pixels on the R-plane. If the R-plane is not enough to accommodate all secret bits, then the process continues in the same way on the G and B planes (Green and Blue planes). If it is still not enough, repeat the procedure above replacing the 2nd LSBs of the pixels on all RGB planes, and continue replacing the 3rd LSBs if the 2nd LSBs of the RGB plane are enough to accommodate the permuted sequence. The process terminated when all secret bits are concealed within the pixels of the cover image, subject to the constraint that the number of modified LSBs should be not more than 3.

For example, to hide the permuted secret binary sequence "**01100101**" into a BMP 24-bit color image, take eight consecutive pixels starting from top left corner of the R-plane of the cover image. Assume the equivalent binary bit pattern of those pixels look-like:

Pixel Value (Decimal)	39	233	200	39	200	233	200	39
Pixel Value (Binary)	00100111	11101001	11001000	00100111	11001000	11101001	11001000	00100111

Figure 3. Pixel values of a cover image.

Then each bit of permuted sequence bits "**01100101"** are copied sequentially to the LSB's of the pixels, resulting in the following bit pattern:

Pixel Value (Decimal)	38	233	201	38	200	233	200	39
Pixel Value (Binary)	0010011**0**	1110100**1**	1100100**1**	0010011**0**	1100100**0**	1110100**1**	1100100**0**	0010011**1**

Figure 4. Pixels values of a stego image.

3.2.1. Stego header

As it can be seen from the above discussion that there are some information are required to enable successful data recovery from the stego image; these include:

(1) Key or password to generate the seed for the RNG (*K*).
(2) Block or permutation size (*N*)
(3) Number of blocks (*B*)
(4) Stego Header Length (SHL)

All of these information and few others are stored in the stego header. Thus, the actual embedded binary sequence comprises the stego header and the permuted secret binary sequence. Figure 5

shows the structure of the stego header, which consists of 9 fields. The names, lengths, and descriptions of these fields are summarized in Table 2.

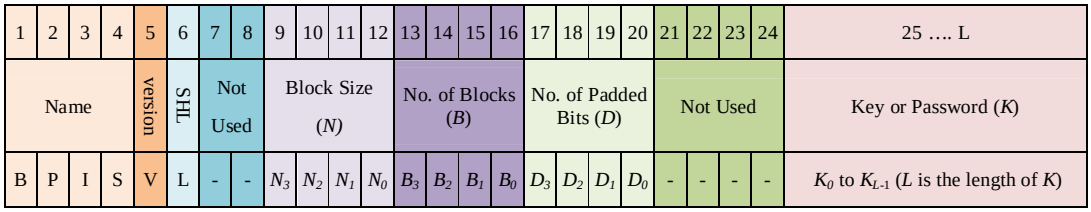


Figure 4. Structure of the stego header.

Table 2. Fields of the stego header.

#	Field Name	Length (Byte)	Description
1	Name	4	Algorithm name which is fixed to BPIS
2	Version	1	Algorithm version
3	SHL	1	Stego header length
4	Not-Used	2	Not used to accommodate future evolution
5	Block Size (*N*)	4	Block size (Maximum of 65536 Byte)
6	No. of Blocks (*B*)	4	No. of block allowing to a maximum of 65536 blocks
7	No. of Padded bits (*D*)	4	No. of padded bits (Maximum of 65536 bits)
8	Not-Used	4	Not-used space for future evolution.
9	Encrypted Key (*K*)	*L*	Encrypted key or password

3.2.2. Algorithm capacity

In this work, since we allow up to a maximum of 3 LSBs to be replaced out of the 8-bit plane color, then maximum capacity of the algorithm is 37.5%. Therefore, in order to ensure hiding the whole secret message into a cover image, the relationship between the size of the cover image (*S*) and the size of the secret message (*M*) is:

$$8 \cdot (M + H) + D \leq 3 \cdot (S – 54) \quad (2)$$

Or $$8 \cdot (M + L + 24) + D \leq 3 \cdot (S – 54) \quad (3)$$

Where *M*, *H*, *L*, and *S* are the secret message size, header length, length of the password or key, and image size (All are in Byte), and *D* is the number of padded bits (Bit). Thus, *S* and *M* should always satisfy the following equation:

$$S \geq 54 + \frac{8(M+L+24)+D}{3} \quad (4)$$

$$M \leq \left[\frac{3(S-54)-D}{8}\right] - (L + 24) \quad (5)$$

3.2.3. Securing stego key (*K*)

In order to secure *K*, the key or password used for generating the seed for the RNG, it is encrypted and embedded within the stego image as part of the stego header. *K* can be secured using one of the following approaches.

(1) Encrypt *K* using a symmetric encryption algorithm (e.g., DES, 3DES, or AES) [13]. In this case, the communicating parties should agree on a certain procedure to exchange the

symmetric encryption key (K_{sym}) securely. This can be accomplished as follows: Encrypt K_{sym} using public-encryption algorithm (e.g., RSA), in other words encrypt K_{sym} using the public-key of the recipient; so that the recipient can decrypt the encrypted K_{sym} using his private-key, reconstruct K, and then proceed with the next secret message recovery steps.

(2) Encrypt K using public-encryption algorithm (e.g., RSA) [13]. in other words encrypt K using the public-key of the recipient; so that the recipient can reconstruct K using his private-key, and then proceed with the next secret message recovery steps.

3.3. Performance measures

The performance of the BPIS algorithm is evaluated by estimating two interrelated parameters, namely, the Mean Square Error (MSE) and the Peak Signal to Noise Ratio (PSNR). In this case, the MSE is the cumulative squared error between the stego and cover images, and it is expressed as [19, 22]:

$$MSE = \frac{1}{X \cdot Y} \sum_{y=1}^{Y} \sum_{x=1}^{X} [I(x,y) - K(x,y)]^2 \quad (6)$$

Where $I(x,y)$ and $K(x,y)$ represent the value of the pixels at position x,y on the cover and stego images respectively, and X and Y represent the dimensions of the images. The PSNR is a measure of quality variation between the cover and stego images, and it is usually expressed in terms of the logarithmic decibel (dB) scale as follows [19, 22]:

$$PSNR = 20 \cdot log_{10} \left(\frac{MAX_I}{\sqrt{MSE}} \right) \quad (7)$$

Where, MAX_I is the maximum possible pixel value of the image. When the pixels are represented using 8-bit per pixel, this it is 255. More generally, when samples are represented using linear Pulse Code Modulation (PCM) with n-bit per pixel, MAX_I is 2^n-1. For color images with three RGB values per pixel, the definition of PSNR is the same except the MSE is the sum over all squared value differences divided by image size and by three. The larger PSNR indicates only little difference is existed between the cover and stego images, which makes it difficult for the attacker to discover the existence of covered message inside the image.

4. RESULTS AND DISCUSSIONS

In order to investigate the performance of the BPIS algorithm, the algorithm is used in a number of experiments to hide text files of different sizes into various BMP image files, and the effect of hiding the text file on stego image quality expressed in terms of PSNR for each steganography experiment is estimated. In particular, in this paper, four text files of different sizes are selected from a standard text files corpus, namely, the Calgary corpus [29, 30].

The text files chosen and their sizes are: paper5 (11954 Byte), paper1 (53161 Byte), paper2 (82199 Byte), and Bib (111261 Byte). The BMP image file selected in this investigation is Flowers.BMP, from which we create three image files of different sizes as summarized in Table 3. The table also shows the maximum allowable capacity for each image for three LSB bit capacities, 1, 2, and 3 bit per pixel (bpp).

Table 3. BMP image files

#	Image Filename	Size	Dimensions (Pixels)	Maximum Capacity (C_{max}) (Byte)		
				1-bpp	2-bpp	3-bpp
1	Flowers-Small	346,854 Byte (339 KB)	400x289	43350 (42 KB)	86700 (84 KB)	130050 (127 KB)
2	Flowers-Medium	439,454 Byte (430 KB)	450x325	54843 (53 KB)	109686 (107 KB)	164529 (160 KB)
3	Flowers-Large	543,054 Byte (531 KB)	500x362	67875 (66 KB)	135750 (132 KB)	203625 (198 KB)

Other input data includes: the length of the permutation P is 128 (i.e., N=128), and the password or the key used as a seed for the RNG is "computer". The results obtained are given in Table 4, which show following:

(1) The algorithm maintains high stego image quality as the PSNR values are always above 30 dB, which means only slight variations are introduced in pixel values in comparison with pixel values of the cover image.
(2) If the same text file is concealed in image files of various sizes, the PSNR of the stego image increases with increasing image file size. This is because as the image size increases the numbers of pixels that may suffer variations in their values are less. For the same reason, using the same image file as a cover image, the PSNR of the stego image increases as the size of the text file decreases. In general, the PSNR is inversely proportional to R, where R is the ratio between the size of the text file (secret message) and the maximum image capacity (C_{max}), i.e., $R=S_{sec}/C_{max}$). For example, the PNSR=40 dB for R=85.6%, and 58 dB for R=5.9%.
(3) Furthermore, in order to realize the effect of introducing permutation on the stego image quality, the PSNR of the stego image is estimated for non-permutated data concealment. The PSNRs show almost a 100% agreement with their equivalent PSNRs.

Table 4. Experimental results.

#	Text File	Flowers-Small			Flowers-Medium			Flowers-Large		
		R (%)	PSNR (dB)		R (%)	PSNR (dB)		R (%)	PSNR (dB)	
			No perm.	BPIS		No perm.	BPIS		No perm.	BPIS
1	Paper5	9.2	56.775	56.740	7.3	57.761	57.771	5.9	58.692	58.716
2	Paper1	40.9	48.920	48.907	32.3	51.278	51.269	26.1	52.211	52.220
3	Paper2	63.2	45.469	45.472	50.0	47.168	47.174	40.4	49.032	49.040
4	Bib	85.6	40.824	40.818	67.6	44.754	44.754	54.7	46.508	46.499

5. CONCLUSIONS

This paper presented a description of a new secure block permutation image steganography (BPIS) algorithm. The algorithm amalgamates the simple concept of permutation with steganography to provide a relatively fast, reliable, and high information security approach. Assuming, the steganalysis succeeded in retrieving or predicting the stego bits, and length of the permutation N, then he/she still has to work hard to predict the actual permutation, where using brute-force attack the complexity could require N! trials.

It is highly recommended to carry on further evaluation for this algorithm, such as: investigating the effect of the permutation size on the algorithm performance in terms of the effect on the stego image quality and processing time. Investigate the performance of the algorithm for concealing other types of secret messages, e.g., audio files, smaller images, other types of data file formats. Finally, uses other image files as cover images.

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] S. Katzenbeisser and F. A. P. Petitcolas. Information Hiding Techniques for Steganography and Digital Watermarking. Artech House Inc., 2000.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] Amin Hashemi Pour and Ali Payandeh. A New Steganography Method Based on the Complex Pixels. Journal of Information Security (JIS), Vol. 3, No. 3, pp. 202-208, 2012.

[10] 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.

[11] 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.

[12] 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.

[13] H. Sajedi, M. Jamzad. A Steganalysis Method based on Contourlet Transform Coefficients. Proceedings of the International Conference of Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP'08), pp. 245-248, Harbin, China, 15-17 August, 2008.

[14] William Stallings. Cryptography and Network Security Principles and Practices, 5th Ed., Prentice-Hall, 2010.

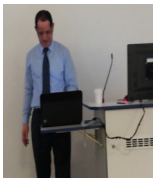
[15] Nameer N. El-Emam, Rasheed Abdul Shaheed Al-Zubidy. New Steganography Algorithm to Conceal a Large Amount of Secret Message Using Hybrid Adaptive Neural Networks with Modified Adaptive Genetic Algorithm. Journal of Systems and Software, Vol. 86, Issue 6, pp. 1465-1481, June 2013.

[16] 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.

[17] 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.

[18] 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.
[19] 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.
[21] 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.
[22] 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.
[23] H. Farid. Detecting Hidden Messages Using Higher-Order Statistical Models. Proceedings of IEEE International Conference on Image processing, Vol. 2, pp. 905-908, New York, USA, 2002.
[24] S. W. Lyu and H. Farid. Steganalysis Using Higher-Order Image Statistics. IEEE Transaction on Information Forensic Security, vol. 1, pp. 111-119, 2006.
[25] 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, June 2010.
[26] I. Avcibas, N. D. Memon, and B. Sankur. Steganalysis of Watermarking Techniques Using Image Quality Metrics. Proceedings of SPIE Security and Watermarking of Multimedia Content III, Vol. 4314. Pp. 523-531, 2001.
[27] I. Avcibas N. D. Memon, and B. Sankur. Steganalysis Using Image Quality Metrics. IEEE Transaction on Image Process, Vol. 12, pp. 221-229, 2003.
[28] H. Sajedi and M. Jamzad. ContSteg: Contourlet-Based Steganography Method. Journal of Wireless Sensor Network, Scientific Research Publishing (SRP), California, US, Vol. 1, No. 3, pp. 163-170, 2009.
[29] T. C. Bell, J. C. Cleary, and I. H. Witten. Text Compression. Prentice-Hall, 1990.
[30] Text Corpora. Retrieved from http://corpus.canterbury.ac.nz/descriptions/#calgary on 28 July 2013.

Author



Hussein Al-Bahadili (hbahadili@uop.edu.jo) is an associate professor at University of Petra (Amman, Jordan). He received his PhD and M.Sc degrees from Queen Mary College, University of London (London, UK) in 1991 and 1988. He received his B.Sc in Engineering from University of Baghdad (Baghdad, Iraq) in 1986. He is a visiting researcher at the Centre of Wireless Networks and Communications (WNCC) at the School of Engineering, University of Brunel (UK). He has published many papers in different fields of science and engineering in numerous leading scholarly and practitioner journals, and presented at leading world-level scholarly conferences. He is the editor of a book titled Simulation in Computer Network Design and Modeling: Use and Analysi. He has published more than ten chapters in a book in prestigious books in IT. His research interests include computer networks design and architecture, routing protocols optimizations, parallel and distributed computing, cryptography and network security, data compression, software and Web engineering.