Indonesian Journal of Electrical Engineering and Computer Science
Vol. 3, No. 3, September 2016, pp. 586 ~ 600
DOI: 10.11591/ijeecs.v3.i2.pp586-600

Blind Steganography in Color Images by Double Wavelet Transform and Improved Arnold Transform

Mohammad Rasoul PourArian[1], Ali Hanani*[2,3]
[1]Department of Computer Engineering, Collage of Technical and Engineering, Kermanshah Branch, Islamic Azad University, Kermanshah, Iran
[2]Department of Computer Engineering, Kermanshah Branch, Islamic Azad University, Kermanshah, Iran
[3]Department of Computer Engineering, Songhor and Koliaei Branch, Islamic Azad University, Songhor and Koliaei, Iran
*Corresphonding author, e-mail: Ali.Hanani@iauksh.ac.ir

Abstract

Steganography is a method which can put data into a media without a tangible impact on the cover media. In addition, the hidden data can be extracted with minimal differences. In this paper, two-dimensional discrete wavelet transform is used for steganography in 24-bit color images. This steganography is of blind type that has no need for original images to extract the secret image. In this algorithm, by the help of a structural similarity and a two-dimensional correlation coefficient, it is tried to select part of sub-band cover image instead of embedding location. These sub-bands are obtained by 3-levels of applying the DWT. Also to increase the steganography resistance against cropping or insert visible watermark, two channels of color image is used simultaneously. In order to raise the security, an encryption algorithm based on Arnold transform was also added to the steganography operation. Because diversity of chaos scenarios is limited in Arnold transform, it could be improved by its mirror in order to increase the diversity of key. Additionally, an ability is added to encryption algorithm that can still maintain its efficiency against image crop. Transparency of steganography image is measured by the peak signal-to-noise ratio that indicates the adequate transparency of steganography process. Extracted image similarity is also measured by two-dimensional correlation coefficient with more than 99% similarity. Moreover, steganography resistance against increasing and decreasing brightness and contrast, lossy compression, cropping image, changing scale and adding noise is acceptable

Keywords: *steganography, discrete wavelet transform arnold transform*

1. Introduction

Steganography is a Greek word consisted of two parts: "Steganos" meaning "cover" and "Graptos" meaning "writing". The aim of steganography is to conceal data in a way that only sender and receiver be aware of the communication and the information [1]. In steganography a set of ordinary and routine data are used such as images, audios, videos and texts [2].

Steganography makes it possible for news and information to be sent to the destination without being controlled, inspected and censored and also with no fear of tracking. Steganography can be used to carry out secret exchanges so that governments and organizations might utilize it for different purposes. While an encrypted message is being transmitted, eavesdroppers realize that important data are being transmitted but steganography makes it possible for substantial data to pass against eavesdroppers without them realizing the transmitted data are of importance [3].

In transform domain, steganography is implemented in various ways. Some of these transforms include Discrete Cosine Transform (DCT), Discrete Fourier Transform (DFT), Discrete Wavelet Transform (DWT), contourlet transform, curvelet transform and ridgelet transform.

1.1. Blind Method in Steganography

Three methods have been proposed for extracting secret image from cover image:

a) non blind method
b) semi blind method
c) blind method

Received April 2, 2016; Revised July 25, 2016; Accepted August 10, 2016

In non blind and semi blind methods, when extracting the secret image, it is supposed that there is access to the cover image or some parts of it. In blind steganography, there is only access to the stego image. In non blind method, the intended part could be obtained from the cover image. Non blind and semi blind methods can be utilized in watermarking subject.

One of the main differences between watermarking and steganography is that there is no particular sender and receiver in watermarking and both embedder and extractor are available to image's owner. In fact, image's owner embodies his watermark invisibly and makes it available to others. Whenever needed, he extracts his watermark to prove to others who is the owner of the image. But in steganography image's ownership is not considered. Images are totally ordinary and they are used as a cover to hide an important image within themselves. Secret Image is of great importance and they should not be recognizable at all by unauthorized people. Embedder is provided to the sender and extractor to the receiver. Secret image is hided in the cover image and transmitted to the destination through communication channels such as internet or it is put in a place where receiver can download the stego image and then extract it.

1.2. Wavelet Transform

Wavelets are mathematical functions used to decompose signals to their frequency components and dimensions of each component are equal to its scale. Wavelet transform is function decomposition based on wavelet functions. There are two types of wavelet transform: Continuous Wavelet Transform (CWT) and discrete wavelet transform [4]. One of the most famous discrete wavelet functions is Haar Wavelet. This function was introduced by the Hungarian mathematician Alfred Haar [5]. Discrete wavelet transform can be implemented based on different functions which are namely called filter. Since images have two dimensions, discrete wavelet transform should be performed two times which is called Two Dimensional Discrete Wavelet Transform (2-D DWT). The procedure is that DWT is first applied on rows and then on columns of the image. Two dimensional discrete wavelet transform divides the image into four different frequency bands. LL band contains information of lower frequency (with the help of low-pass filter), HL and LH bands includes the information of middle frequencies (with the help of low-pass and high-pass filters), and HH band contains the information of high frequency (with the help of high-pass filter). LL band is the approximation of the image and HL, LH and HH bands respectively represent horizontal, vertical and diagonal edges. The index written next to them defines the decomposition levels [6].

Figures 1 and 2 shows two-dimensional discrete wavelet transform process.

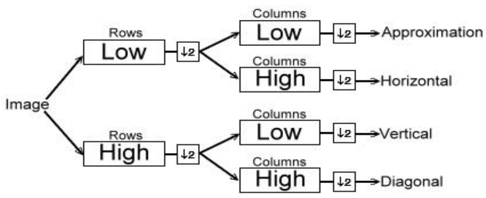

Figure 1. Low-Pass & High-Pass filters in DWT

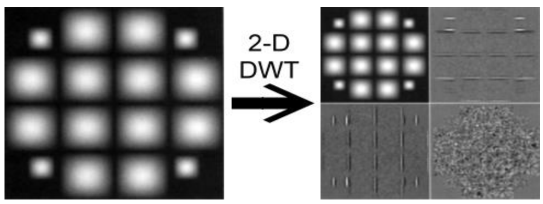

Figure 2. 4-Bands in 1-Level DWT

1.3. Related work

To review related work, many articles were studied. All reviewed algorithms in this study perform steganography through blind method. Some of these articles are briefly explained.

In [7-9] and [10] a pseudo random signals string or numeric sequence is used as secret data. There are no secret images in this methods and the aim of extraction process is to determine whether there is secret data in the image or not.

In [11-13] and [14] secret images are changed to binary mode and their size is often small. In some articles, steganography in place domain has helped steganography in transform domain so that secret image bits in Least Significant Bit (LSB) are replaced with certain bits of frequency coefficients. These algorithms have generally used DWT, Discrete Wavelet Packet Transform (DWPT), DCT, Singular Value Decomposition (SVD) and other methods of transform domain.

Zhao in [15] has concealed a 64×64 binary image in a 512×512 color image through DCT and DWT. He has used Arnold transform for secret image encryption to make the image chaotic. In this article, it has been tried to overcome the problem of high noises that might be placed on the image like a visible watermark. Two images are extracted in extraction process, one is introduces as semi fragile image and the other as tamper image thus damaged parts of the first image are available in the second image and vice versa.

In [16], Kalra has concealed a 32×32 binary image in a 256×256 8-bit image through DWT. In this article, Arnold transform is first used to increase encryption security then logistic function is used for achieving higher security.

Sarker in [17] has used a 64×64 binary image as the secret image and a 512×512 8-bit image as the cover image; he has also utilized Arnold transform for encryption. Then, he transforms secret and cover images to Hadamard matrices using Hadamard transform. He performs steganography processes in Hadamard space and at last gets help from Hadamard reverse.

Yongqi in [18] has concealed a 64×64 binary image in a 512×512 color image. In this article, Integer Wavelet Transform (IWT) is used instead of DWT. Its encryption has been performed by logistic function.

In [19], Atawneh has used blind steganography as a combination based DWT and LSB. First, secret image is encrypted using Arnold transform, and then its bits is being Exclusive OR (XOR) with some bits of the cover image. These bits are selected from a certain part of the cover image the coordinates of which are changeable and are considered as two stego keys. Finally, the obtained code map is concealed in LL and LH bands. So, three stego keys are used in this algorithm. First key is related to Arnold transform, second and third keys are related to the intended pixel coordinates of the cover image. This way, low security problem is being overcome by second and third stego keys additions. According to the results obtained from this article, this algorithm is somehow able to withstand lossy compression and some image processing attacks but it's not robust to geometric attacks such as rotate and crop.

One of the advantages of our steganography method is that a 8-bit image can be concealed in a 8-bit channel in a way that there would be no need for primary images. Converting images from 8-bit to 2-bit diminishes image details because binary images only use two colors: black and white. Figure 3 illustrates some examples of a 8-bit image transform to a 2-bit image.

Hiding a colored image in another colored image is one of main advantages of our suggested method. The problem faced in the previous methods was that the secret image was 2-bit. Having such steganography in transform domain is necessary. We are seeking for providing you with a method which not only enjoys the above mentioned advantage, but also has a proper security, robustness and transparency.

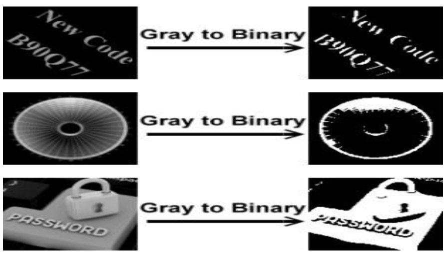

Figure 3. Samples of Convert Grayscale to Binary

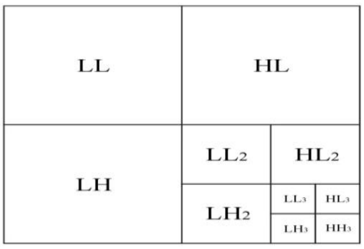

Figure 4. Decompose 3-Levels DWT based on High-Pass filter

In section 2 we will describe our proposed algorithms for steganography and encryption and we will also provide some explanations on Arnold Transform. Section 3 investigates and compares the performance of proposed algorithms and finally, section 4 concludes.

2. Research Method
2.1. The Proposed Algorithm for Steganography

Our steganography algorithm has been implemented based on 2-D DWT with Haar filter. We have used two dimensional discrete wavelet transform in three levels. In the first level, the image is decomposed to four LL, HL, LH and HH bands. In the second level, the HH band is decomposed to four sub-bands: LL_2,HL_2,LH_2 and HH_2 and in the third level the HH_2 sub-band is decompose to four sub-bands: LL_3,HL_3,$LH_3$and HH_3. In each level, length and width of the image are halved. We used three level DWT to be able to better conceal the secret image because this way, lesser parts of cover image are changed. Figure 4 illustrates frequencies analysis.

For example if original dimensions are 800×800 pixels, LL, HL, LH and HH bands would have 400×400 pixels dimensions. In the next level, LL_2,HL_2,LH_2 and HH_2 sub-bands would have 200×200 pixels dimensions, and then in the final level, LL_3,HL_3‹LH_3 and HH_3 sub-bands would have an area of 100×100 pixels. Generally three equations can be used to integrate two images.

$$v' = v + \alpha x \quad (1)$$

$$v' = v + v\alpha x \quad (2)$$

$$v' = v \exp(\alpha x) \quad (3)$$

Equation (1) is known as increased integration and we have used it in our proposed algorithm. In this equation, v is the cover image and x is the secret image. The amount of alpha (α) is a fraction number multiply by the secret image to decrease its values. This number is selected arbitrary. The value of alpha can be determined through trial and error, but a set of equations can be used for obtaining the best value of alpha. This equations measure Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity (SSIM) so they can be helpful in obtaining the accurate value of alpha. If the value of alpha equals 1, it is the secret image and as a result, noise or distortion would appear on the cover image. These noises can be seen as dots on the cover image so the nature of steganography would face troubles. If the value of alpha equals zero, the secret image would be completely destroyed and would not be extractable. So the value of alpha should be a fraction number between 0 to 1.

The value of alpha should be determined carefully since there is compromise between two robustness and transparency criteria meaning as the robustness increases, transparency decreases and vice versa. We cannot have the highest transparency and robustness together so there should be a balance between them. In the proposed algorithm, we have selected alpha as 0.25 through trial and error in various conditions.

Next issue is the selection of integration place meaning the secret image should be exactly concealed in which part of the cover image. In DWT-based algorithms, secret image is concealed in one of sub-bands. In our proposed algorithm, we selected HH_3 sub-band as the placement part. In doing so, we decomposed the HH band to reach HH_2 sub-band, then HH_2 sub-band was decomposed for obtaining HH_3 sub-band. If $LL \rightarrow LL_2 \rightarrow LL_3$ path be used instead of $HH \rightarrow HH_2 \rightarrow HH_3$ path, secret image would be a part of cover image which has a negative effect on visual quality of the cover image. To solve this problem, the value of alpha should be decreased to the point that secret image pixels disappear from the cover image. It should be noted that pixels values in simulator programs such as MATLAB can be integer values or fraction values (single and double) but when the image is saved as common formats like BMP, the value of each pixel is converted to an integer between 0 to 255. Sub-bands taken from DWT are all double, so secret image should be converted to double before integration. In order to reduce the negative effects of distortions and noises on cover image, steganography in LL_3,HL_3 and LH_3 sub-bands requires smaller alpha compared to steganography in HH_3 sub-band. The quality of extracted image reduces as the value of alpha decreases. As long as these algorithms exist in simulator program environment and there is the possibility of using fraction values, secret image would be extracted with the exact same initial accuracy. Compared to steganography in LL_3,HL_3 and LH_3 sub-bands, steganography in HH_3 sub-band has robust when applying the median filter. Median filter is one of image processing attacks which removes high frequencies (H). Steganography in LL_3 is largely robust to this attack, HL_3 and LH_3 also, show some degree of robustness but steganography in HH_3 sub-band is not robust to this attack. With all these descriptions, we used HH_3 sub-band in our proposed algorithm because human eyes are less sensitive to noises appears in high frequencies.

There are two important issues in choosing the placement location:

a) The cover image should have no distortions
b) The quality of the extracted image have an acceptable level

Up to here, we could determine alpha and the place of steganography.

This process is shown in equation 4:

$$HH_3' = HH_3 + (0.25 \times Secret) \quad (4)$$

In this equation, the value of HH_3' is that of HH_3 to which the secret image has been added.

Extraction method is shown in equation 5:

$$Secret = (HH_3' - HH_3) \div 0.25 \quad (5)$$

We should be able to obtain the same secret image from the Equation (5), meaning that the value of the Secret is unknown in this equation. Value of HH_3' can be measured by three levels decomposition of stego image and we know that the amount of alpha is 0.25. Here, the problem is that the value of HH_3 is unknown as well because it has been integrated with the secret image and we do not know its value before integration.

Our proposed method is that some needed parts of the cover image be put in the stego image itself. But the question is that this part of image (here HH_3) should be put in which part of the image that does not damage the cover image. The answer is that we should search for a sub-band that has the same size as the intended part and is visually the most similar one to it.

If we decompose LL band twice we would have:

$(LL \rightarrow LL_2 \rightarrow LL_3)$, $(LL \rightarrow LL_2 \rightarrow HL_3)$, $(LL \rightarrow LL_2 \rightarrow LH_3)$, $(LL \rightarrow LL_2 \rightarrow HH_3)$,
$(LL \rightarrow HL_2 \rightarrow LL_3)$, $(LL \rightarrow HL_2 \rightarrow HL_3)$, $(LL \rightarrow HL_2 \rightarrow LH_3)$, $(LL \rightarrow HL_2 \rightarrow HH_3)$,
$(LL \rightarrow LH_2 \rightarrow LL_3)$, $(LL \rightarrow LH_2 \rightarrow HL_3)$, $(LL \rightarrow LH_2 \rightarrow LH_3)$, $(LL \rightarrow LH_2 \rightarrow HH_3)$,
$(LL \rightarrow HH_2 \rightarrow LL_3)$, $(LL \rightarrow HH_2 \rightarrow HL_3)$, (LL→HH2→LH3), (LL→HH2→HH3).

If we decompose HL band twice we would have:

(HL→LL2→LL3), (HL→LL2→HL3), (HL→LL2→LH3), (HL→LL2→HH3),
$(HL \rightarrow HL_2 \rightarrow LL_3)$, $(HL \rightarrow HL_2 \rightarrow HL_3)$, $(HL \rightarrow HL_2 \rightarrow LH_3)$, $(HL \rightarrow HL_2 \rightarrow HH_3)$,
$(HL \rightarrow LH_2 \rightarrow LL_3)$, $(HL \rightarrow LH_2 \rightarrow HL_3)$, $(HL \rightarrow LH_2 \rightarrow LH_3)$, $(HL \rightarrow LH_2 \rightarrow HH_3)$,
$(HL \rightarrow HH_2 \rightarrow LL_3)$, $(HL \rightarrow HH_2 \rightarrow$HL3), (HL→HH2→LH3), (HL→HH2→HH3).

If we decompose LH band twice we would have:

$(LH \rightarrow LL_2 \rightarrow LL_3)$, $(LH \rightarrow LL_2 \rightarrow HL_3)$, $(LH \rightarrow LL_2 \rightarrow LH_3)$, $(LH \rightarrow LL_2 \rightarrow HH_3)$,
$(LH \rightarrow HL_2 \rightarrow LL_3)$, $(LH \rightarrow HL_2 \rightarrow HL_3)$, $(LH \rightarrow HL_2 \rightarrow LH_3)$, $(LH \rightarrow HL_2 \rightarrow HH_3)$,
$(LH \rightarrow LH_2 \rightarrow LL_3)$, $(LH \rightarrow LH_2 \rightarrow HL_3)$, $(LH \rightarrow LH_2 \rightarrow LH_3)$, $(LH \rightarrow LH_2 \rightarrow HH_3)$,
$(LH \rightarrow HH_2 \rightarrow LL_3)$, $(LH \rightarrow HH_2 \rightarrow$HL3), (LH→HH2→LH3), (LH→HH2→HH3).

If we decompose HH band twice we would have:

$(HH \rightarrow LL_2 \rightarrow LL_3)$, $(HH \rightarrow LL_2 \rightarrow HL_3)$, $(HH \rightarrow LL_2 \rightarrow LH_3)$, $(HH \rightarrow LL_2 \rightarrow HH_3)$,
$(HH \rightarrow HL_2 \rightarrow LL_3)$, $(HH \rightarrow HL_2 \rightarrow HL_3)$, $(HH \rightarrow HL_2 \rightarrow LH_3)$, $(HH \rightarrow HL_2 \rightarrow HH_3)$,
$(HH \rightarrow LH_2 \rightarrow LL_3)$, $(HH \rightarrow LH_2 \rightarrow HL_3)$, $(HH \rightarrow LH_2 \rightarrow LH_3)$, $(HH \rightarrow LH_2 \rightarrow HH_3)$,
$(HH \rightarrow HH_2 \rightarrow LL_3)$, $(HH \rightarrow HH_2 \rightarrow HL_3)$, $(HH \rightarrow HH_2 \rightarrow LH_3)$, $(HH \rightarrow HH_2 \rightarrow HH_3)$.

By decomposing each band, 16 sub-bands are created that totally form 64 sub-bands. Now, we have to examine which of these sub-bands are more similar to $HH \rightarrow HH_2 \rightarrow HH_3$ sub-band. All 64 sub-bands are of the same size, but their content isn't the same and they are visually different. In order to examine the similarities, two PSNR and SSIM indexes were used.

By studying these indexes in dozen of images, we concluded that these 6 sub- bands are the most similar to $HH \rightarrow HH_2 \rightarrow HH_3$ sub-band:

($HH \rightarrow LL_2 \rightarrow HH_3$), ($HH \rightarrow HL_2 \rightarrow HH_3$), ($HH \rightarrow LH_2 \rightarrow HH_3$),
($HH \rightarrow HH_2 \rightarrow LL_3$), ($HH \rightarrow HH_2 \rightarrow HL_3$), ($HH \rightarrow HH_2 \rightarrow LH_3$).

Generally, these 6 sub-bands can be used and we used $HH \rightarrow HH_2 \rightarrow HL_3$ sub-band. Discrete wavelet transform should be inversed after integration. The inverse of DWT is shown as IDWT (Inverse Discrete Wavelet Transform). Similar to DWT, IDWT is performed in three levels so that the image would return from frequency state to the initial state. Finally, the integration algorithm would be as Equation (6):

$$HH_3=HL_3+ (0.25\times Secret) \tag{6}$$

It means instead of adding HH_3 sub-band to the secret image, we used HL_3 sub- band and its product was placed at the location related to HH_3 sub-band. Final algorithm of image extraction is shown in Equation (7):

$$Secret= (HH_3-HL_3) \div 0.25 \tag{7}$$

Here, first the stego image is decomposed in three levels using DWT. Then HL_3 sub-band is subtracted from HH_3 sub-band which contains data. At the end, the difference is divided by alpha value to extract the secret image. Figure 5 illustrates the placement process of the secret image in HH_3 place.

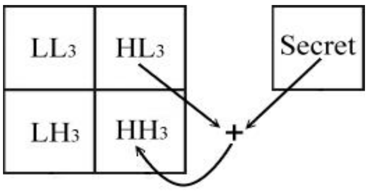

Figure 5. Embedding in HH_3 Sub-Band

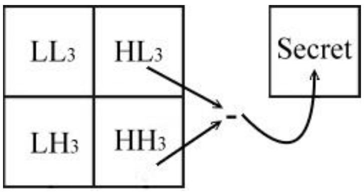

Figure 6. Extracting from HH_3 Sub-Band

Figure 6 illustrates the extraction process of the secret image from HH_3 place with the help of LH_3.

This steganography method has an acceptable level of robustness to the increase and decrease of brightness and contrast, JPEG compression, upscale, salty and pepper noise and Gaussian noise but it is not robust to image crop and if any part of the cover image is cropped the stego image is cropped as well. Since nowadays images are RGB color images with bit depth of 24-bit, we decided to extend steganography to color images. As you may know, RGB images are consisted of three red, green and blue channels. Each of these channels is exactly a grayscale image with bit depth of 8-bit and its pixels have a value between 0 to 255. To perform steganography in RGB images, first, the three image channels should be separated from each other and then steganography shall be performed on the intended channel. In the proposed method, we aim to conceal the secret image which is a grayscale, in the red channel. Considering the fact that there are two other green and blue channels available to us, we can exploit them in other ways. We decided to use blue channel to increase the robustness to image crop. But how can be this done?

For steganography in the blue channel, first the image related to the blue channel is rotated 180 degrees till image is upside down. Then exactly similar to steganography in the red channel, secret image should be concealed under HH3 sub-band. After the steganography process is over, the image of the blue channel should be rotated 180 degrees to the initial state. When extracting, first the image related to blue channel should be rotated 180 degrees then similar to red channel, the secret image can be extracted.

What are the advantages of 180 degrees rotation?

This 180 degrees rotation causes the two blue and red channels to overlap each other. It means that if some parts of the image is cropped or altered by a great noise, in most cases the secret image can be properly extracted since the pixels that have been destroyed in the red channel exists in the blue channel and vice versa. Using the proposed method, the two extracted images from two blue and red channels complete each other, and this holds true for image crop as well. For example, if the right side of the cover image is cropped, this crop is applied to both red and blue channels, but in the red channel this cut is at right side of the image and in the blue channel, it is at the left side of the image. So a relative robust is created against the image crop. It must be noted that in two-channel steganography two images are extracted after extraction process, if no noise or crop attacks occur, two extracted images would be the same otherwise, the two extracted images would be different and complete each other.

Up to here, we described the proposed steganography method. Now we want to increase the security of steganography method. Since this article may be studies by different people, the extraction algorithm is a public one. Extraction algorithm publicity is in conflict with the nature of this technique because, by implementing it, all stego images of this method can be extracted thus the security of algorithm is challenged. So we need a solution to personalize the algorithm. Our solution is image encryption. The procedure is that first we encrypt a secret image using a key, then we steganograph the encrypted image. To extract the secret image, first we extract the encrypted image then decrypt the extracted image using the same key. We have designed this key as a numeric code to be easy to use. We will describe our encryption algorithm in section 4.

2.2. Encryption in Steganography

Encryption methods can be used for increasing the security of steganography methods. In this technique the nature of the image is altered and it can be turned to the initial state only by having the special key. These types of methods are called chaotic states. Following are among chaotic methods [20, 21]:

a. Logistic method
b. Tent method
c. Sin method
d. Cubic method
e. Chen method
f. Arnold method
g. Barker method
h. Standard method
i. Kaplan Yorke method
j. Ikeda method

The most important methods among them are chaotic Arnold and logistic methods.

2.3. Arnold Transform

This method was invented by a Russian mathematician named Vladimir Igorevich Arnold. This method is also introduced as Arnold's Cat Map (ACM) since it was first implemented on the image of a cat [22]. Two dimensional Arnold transform can be used for images encryption. The procedure is that the pixels of the image are replaced with one another. According to the repetition loop, in each iteration, the amount of pixels displacement changes in different ways. The user is asked to enter a number as iteration, then the loop is repeated based on the entered number and the image becomes chaotic. This process is shown in equation 8 [22]:

$$\binom{x'}{y'} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} . \binom{x}{y} \times \mathrm{Mod(m)} \quad (8)$$

x and y are the coordinates of the image, Mod means the quotient and m is the length or width of the image. x' and y' are the coordinates of the image after being chaotic.

There are some problems for encryption through Arnold transform:

a. Length and width of the image should be equal
b. Increasing the image size slows down the processes
c. It is inefficient regarding image crop

d. The security of encryption key is low

When margins of an image are cropped, some of its pixels are removed. These removed pixels might be related to the central part of the image and if those pixels are not available, then it is impossible to turn the image to the initial state. Generally, Arnold transform algorithm is invented for encryption of images which have equal length and width. Our proposed algorithm overcomes this problem by inserting a sign in the center of the image.

The most important problem with Arnold transform is the security of its encryption key because after a while, its repetition loop returns to its starting point. For example, if we want to encrypt an image with dimensions of 100×100 pixels, the encrypted image is the same as the unencrypted image after 150 times loop repetition, meaning that after 150 times repetition, it returns to the starting point. So if we encrypt this image with a key value of 20, it can be decrypted by 20, 170, 320, 470, 620 keys. So returning point to the starting state for an image with dimensions of 64×64 is 48, for 128×128 is 96, for 256×256 is 192, also for 25×25 is 50, for 100×100 is 150 and for 150×150 is 300. Figure 7 shows an encrypted image in different iteration using Arnold transform.

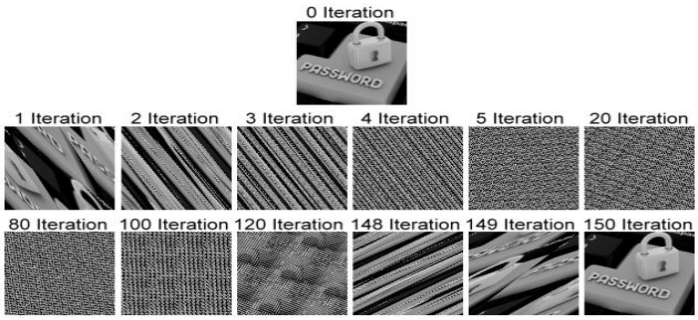

Figure 7. Arnold Transform with Different Iterations

The keys shouldn't be easily obtainable. In Arnold transform, encryption key can be obtained by examining a few states. Thus the security of its encryption key is low. Our proposed algorithm solves the problem of keys limitedness.

2.4. Proposed Algorithm for Encryption

Our proposed algorithm is based on Arnold transform for image encryption. Our aim is to increase chaotic states and therefore eliminate key limitedness. After reviewing and testing, we realized that Arnold transform is not robust to symmetrization. By symmetrization we mean mirror state. It is also called flip horizontally.

For example, if the pixels of an image are like the following matrix: $\begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{matrix}$

Its symmetry would be: $\begin{matrix} 3 & 2 & 1 \\ 6 & 5 & 4 \\ 9 & 8 & 7 \end{matrix}$

As mentioned earlier, Arnold transform has one repetition loop. Now, the chaotic states would increase if we symmetrize the image in some loop repetitions. Accordingly, if we have an array of 0 and 1 values, we can control the algorithm using the array. This means that when 1 is observed the image should be symmetrized and whenever 0 is observed, it should not be symmetrized. As a result, the encryption key would be an n-bit array which is arbitrarily filled with 0 and 1. To encrypt the image, its inverse should be used. Since we wish to deliver the keys to the receiver easily, we converted it from binary state to the base 10 to facilitate key transfer. We make the process clear by providing an example.

If we consider the encryption key as 123456, first it should be converted from base 10 to base 2:

$123456 = (11110001001000000)_2$

The obtained binary number is 17-bit. So Arnold transform is repeated 17 times. Image symmetry is used in first, second, third, fourth, eighth and eleventh. This way the security is so high that even if one of bits changes, image cannot be decrypted. Symmetry of the binary number is required for decryption which is as follows:

$$123456=(11110001001000000)_2 \rightarrow (00000010010001111)_2$$

This numeric code can have a variable length which, in this example, is a six-digit code. Having a variable length makes it impossible for unauthorized people to guess the selected code. Figure 8 shows the result of using wrong codes. The correct password is 583142 and we used wrong passwords with one number lower and higher, it can be seen that the image is not decrypted.

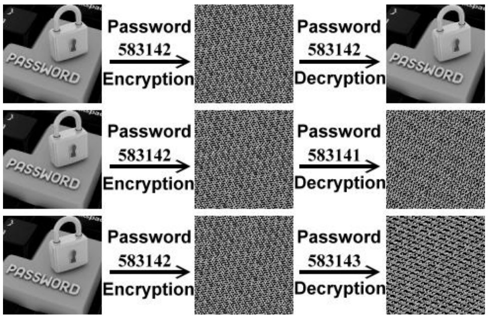

Figure 8. Result of using Correct Password and Wrong Passwords

Figure 9 illustrates the robustness of the proposed algorithm to great noises and visible watermarks. As it can be observed, we located the visible watermark at the corner of the image to examine algorithm's robustness. Visible watermarks have caused observable noises on the extracted image. First image is extracted from the red channel and the second image from the blue channel. As it is obvious, the parts of the red channel image that have noises are intact in the blue channel image and vice versa. This shows the advantage of proposed two-channel algorithm.

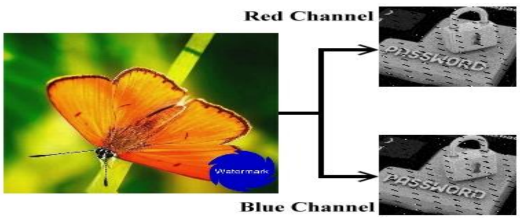

Figure 9. Overlapping of Noisy Pixels by Two-Channels

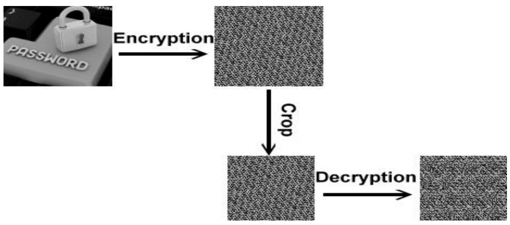

Figure 10. Problem of Arnold Transform in image crop

This encryption has an acceptable level of robustness to the increase and decrease of brightness and contrast, JPEG compression, upscale, salty and pepper noise and Gaussian noise. In the case of applying noises as large spots or inserting visible watermarks, damaged pixels would not be recoverable thus we used two-channel method. It is to some extents robust to raising and lowering the scale but under the condition that image dimensions be turned to the initial state before decryption. It's not robust to rotation unless the image be rotated and turned to the initial state prior to decryption. Also, it's not robust to image crop for which we suggested a solution in section 2-5 so that Arnold transform would be robust to crop.

2.5. Robustness to Image Crop

Imagine that 20 rows are deleted from top of the image and 20 from its left side. What would happen? Figure 10 shows such situation.

Arnold transform aims to move the pixels and put them in their places but some of them are missed and do not exist anymore thus the algorithm is challenged. Our solution to solve this problem is the simulation of removed parts. It means that we place some black parts in the place of removed parts. The problem is that algorithm does not know how much is removed from 4 sides of the image to simulate them. We suppose that the receiver is aware of image's initial size meaning that they have been a pre-agreement between the sender and receiver that, for example, the image is always sent with 200×200 pixels size. The important point with image crop is that the margins of image are removed since central parts of the image are generally more important. Considering this problem, we put a sign at the center of the image. This sign helps us find center of image center even after image's margins crop. We use 5 pixels image to insert this sign, one pixel in center and 4 its neighborhoods. These 4 pixels are located at top, below, left and right sides of the central pixel. Central pixel is recognized with white color (255 value) and 4 neighbor pixels with black color (0 value). Figure 11 illustrates signed pixels with 16 times magnification zoom.

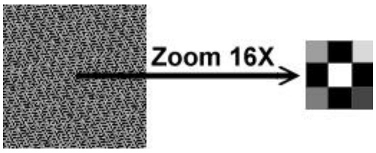

Figure 11. Signed 5-Pixels in Center Of Image

In decryption process, a repetition loop is built with the same area as the image and all pixels would be checked from the beginning. Whenever a white pixel with four pixels in its neighborhood is observed, the repetition loop stops and white pixel is considered as the image center. Now the distance between that point and four sides of the image should be measured. Then removed parts are simulated using black pieces and the cropped image returns to its previous size. Doing so, the problem of image crop is solved and Arnold transform can use black pixels instead of missing ones. This solution causes the decryption process to be easily performed. Figure 12 illustrates this process.

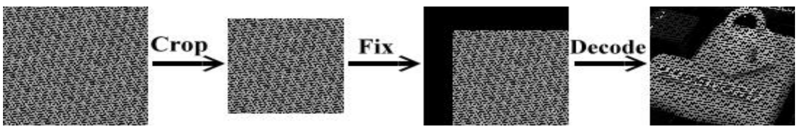

Figure 12. Simulation Lost Pixels in Image Crop

There is a question here. What would happen if at the same time that the image is being cropped, the amount of brightness or contrast change too?

As you know white color equals 255 and black color equals 0. Now if we want to lighten the image 1 unit, the value of all pixels sum to 1, thus the central pixel remains unchangeable with the value 255 but four neighboring pixels alter from 0 to 1 and lighten 1 unit. Also, we will have the opposite of this state for reducing brightness. To solve this problem, algorithm of finding the central point should be a little flexible. It means that instead of obtaining 0 and 255 values, we should search for a defined interval. For example if we want the algorithm to be robust to 30 units increase of brightness, instead of finding 255 number we search for a pixel with a value more than 255 that have four pixels of 0 value in its neighborhood. Algorithm of finding the central point can be more flexible by determining different conditions. It should be noted that if the interval gets too large, then algorithm of finding the central point might encounter problems meaning that it mistakenly finds another point and recognizes it as the central point. It must be mentioned that central point analysis is conducted only when some parts of the image are cropped.

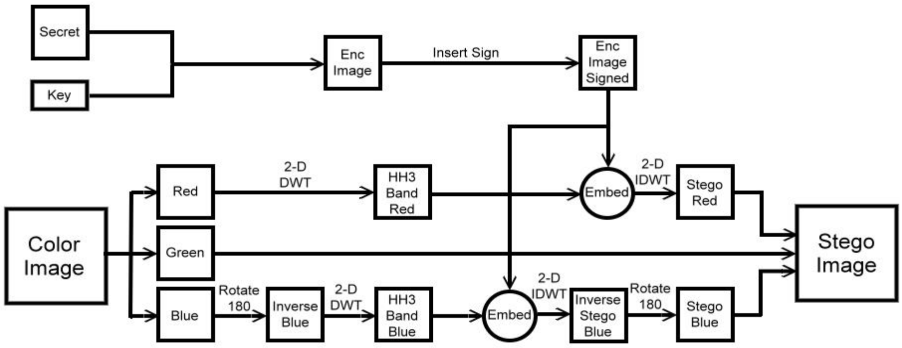

Figure 13. Block Diagram of Embedding with Encryption

Now this encrypted image can be steganographed in a cover image. If our steganography algorithm is made public, it is still personalized due to the encryption algorithm. Because this algorithm is based on a several-digit code that only receiver and sender know about it. As a result, even if other people have the encryption and decryption algorithms, they still can't decrypt the concealed image. Integration and extraction processes are completely illustrated in Figures 13 and 14.

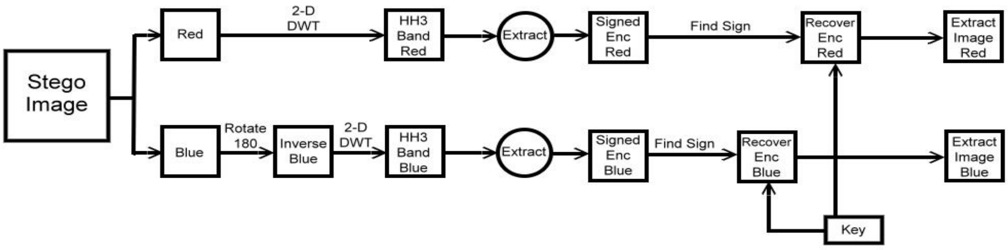

Figure 14. Block Diagram of Extracting with Decryption

3. Results and Analysis

The performance of steganography algorithms can be assessed through different criteria such as Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), Structural Similarity (SSIM) and Normalized Cross Correlation (NCC).

3.1. Performance of Proposed Algorithm

Here, we want to examine the amount of cover image changes using PSNR index. This comparison is performed between the cover image and stego image. In this test, we used 4 secret images and 9 cover images which can be observed in Figures 15 and 16. These images have been saved with BMP format.

Cover images have a size of 800×800 pixels and secret images have a size of 100×100 pixels. The type of cover images is RGB with bit depth of 24-bits and secret images are of grayscale type with bit depth of 8-bits. The implementation is performed through MATLAB 2015 version.

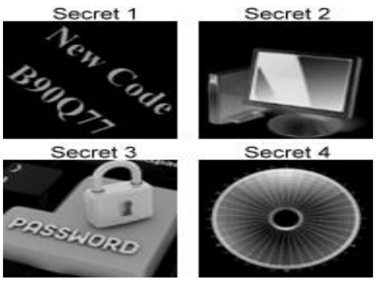

Figure 15. Secret Images

Figure 16. Cover Images

Table 1. PSNR of stego images

	Secret1	Secret2	Secret3	Secret4
Cover1	49.9552	43.2785	41.2180	44.3960
Cover2	49.2540	43.0864	41.1537	44.1958
Cover3	50.5544	43.3682	41.3115	44.5620
Cover4	50.5674	43.3891	41.3070	44.5685
Cover5	50.6245	43.3796	41.3096	44.5772
Cover6	50.2337	43.3047	41.2664	44.4965
Cover7	50.3105	43.3038	41.2704	44.4952
Cover8	50.2375	43.3191	41.2691	44.4757
Cover9	50.4221	43.3459	41.2759	44.5017

Table 2. NCC of Extracted Images

	Secret1	Secret2	Secret3	Secret4
Cover1	0.9943	0.9957	0.9922	0.9933
Cover2	0.9902	0.9949	0.9920	0.9919
Cover3	0.9946	0.9956	0.9920	0.9930
Cover4	0.9941	0.9957	0.9922	0.9931
Cover5	0.9951	0.9958	0.9921	0.9931
Cover6	0.9949	0.9959	0.9922	0.9934
Cover7	0.9914	0.9951	0.9917	0.9924
Cover8	0.9930	0.9955	0.9921	0.9926
Cover9	0.9935	0.9956	0.9921	0.9929

Table 3. NCC of Extracted Images after Attacks

	Attack	NCC
1	Increase Brightness 10%	0.9911
2	Increase Brightness 15%	0.8987
3	Decrease Brightness 10%	0.9919
4	Decrease Brightness 15%	0.9558
5	Increase Contrast 10%	0.9899
6	Increase Contrast 15%	0.8606
7	Decrease Contrast 10%	0.9938
8	Decrease Contrast 15%	0.9937
9	Salty & Pepper Noise, Rate 2%	0.7758
10	Gaussian Noise, Rate 2%	0.6827
11	Median Filter	0.2050
12	JPEG Compression, Rate 1%	0.9919
13	JPEG Compression, Rate 5%	0.9698
14	Crop 640×640	0.8789
15	Rotate 180°	0.8967
16	Downscale to 600×600	0.9330
17	Upscale to 1000×1000	0.9791
18	Gaussian Filter, STD 0.5	0.9793
19	Sharp	0.9771
20	Visible Watermark, Figure 9	0.9305

We embedded each secret image in 9 cover image and each time we recorded the amount of PSNR. The results are shown in Table 1. The values of Table 1 are obtained through single-channel steganography and without applying encryption. By adding encryption process to steganography, the same numbers are obtained with minor differences in decimal. This minor difference is due to the change of the place of image pixels and we cannot consider a fixed number for it because according to the selected password, the arrangement of pixels would be different which has a minor effect on the value of PSNR.

Note: if steganography be performed at the same time as a two-channel method in both blue and red channels, the value of PSNR would be about 3 units lesser than single-channel steganography; however the amount of PSNR would be still acceptable. Now we want to check the quality of extracted images. To do this, we have used two-dimensional correlation. The results shown in Table 2 are obtained from the comparison between the secret image and the extracted image. If encryption process be added to steganography, there would be no negative effect and the exact same numbers would be obtained.

As it can be observed, extracted images have more than 99% conformity with the original image which is an acceptable result.

Up to here, we examined the quality of stego image and extracted image in different scenarios. Now we want to apply some of attacks on the stego image and evaluate the quality of the extracted image. This way we can also examine the robustness of proposed algorithm against intentional and unintentional attacks. Some these attacks include increasing the brightness, decreasing the brightness, increasing the contrast, decreasing the contrast, applying noise, applying filtering, lossy compression, image crop, image rotation, upscale, downscale and sharpening. In applying noise we can name salty and pepper noise and Gaussian noise. In applying filtering we can name median and Gaussian filter. And from Lossy compression we can mention JPEG and JPEG2000 compressions. Building such an steganography algorithm that is able to withstand all these attacks to a high extent is almost impossible. Increasing image robustness in algorithms that are based on non blind and semi blind methods is easier than in steganography algorithms which are based on blind method since there are some parts of images always available to them which are not affected by attacks and can empower their algorithms against attacks using intact parts. In our proposed method, this part is concealed within the image itself isn't not influenced by attacks. However we have tried to increase algorithm robustness as much as possible. In this test we concealed the image named Secret1 in the image named Cover1. The correlation before attacks was 0.9943. Table 3 shows the recorded values for correlation after applying attacks.

Images of these attacks are shown in Figure 17.

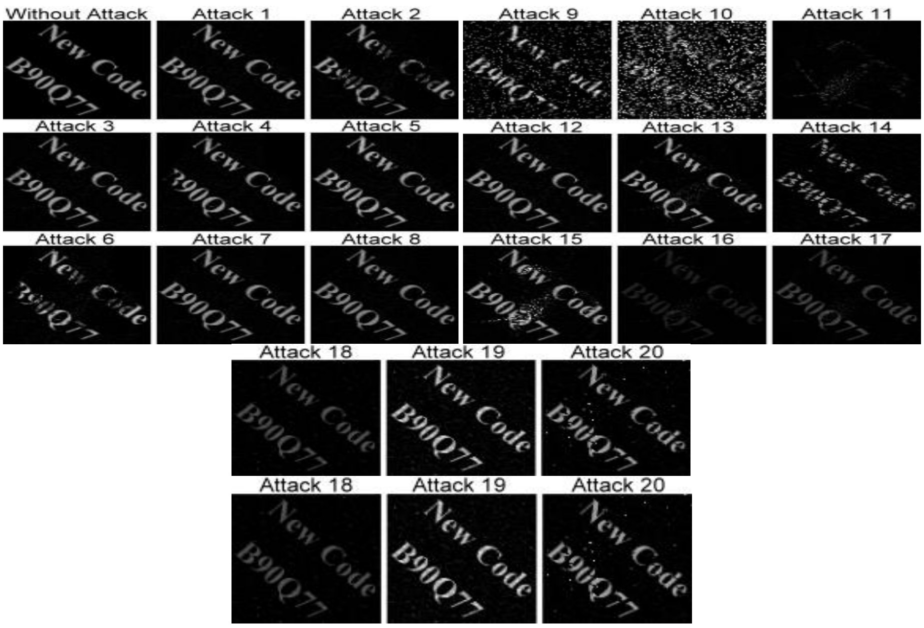

Figure 17. Images Obtained After Attacks

3.2. Comparison of the Proposed Method

Since in most articles, binary images have been used and their secret image is usually a small size logo, it may not be a proper thing to compare those methods with the method proposed in this paper. Because their purpose is to prove image ownership while the aim of this paper is to transfer a secret image to a particular destination. Concealing a watermark logo with bit depth of 2-bit is much easier than concealing a 8-bit image since 2-bit images can be embedded as a binary string. Generally when a small amount of data is concealed, stego image would have more resilience to attacks and as the size and volume of secret data increases it would become more vulnerable to attacks. However obtained value for PSNR is compared to several other articles and provided in Figure 18.

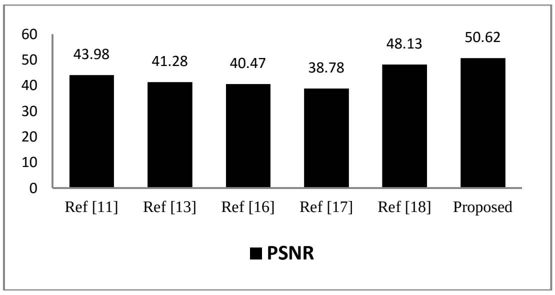

Figure 18. Compare of PSNR Values

Our proposed method has a high level of security and penetrating it is not practically possible while other articles contain no encryption or have implemented Arnold transform with a simple key. For example in [19] pixels coordinates have been used to compensate for the weakness of the main key while Arnold encryption algorithm is not prompted at all. It is obvious that trying to find the coordinates of starting pixel is way easier than guessing a n-digit password.

4. Conclusion

In this article, we embedded a 8-bit image in two 8-bit channels of a color image using two-dimensional discrete wavelet transform. Image of the second channel is 180 degrees rotated prior to steganography so that the extracted images from two channels could overlap each other against image crop and visible watermarks. The two used channels are blue and red channels of the RGB image. Steganography method is blind method which doesn’t require the primary image for extracting the secret image. In order to increase the security of steganography, an encryption algorithm based on Arnold transform was added to steganography process. This encryption algorithm is an improved method of Arnold transform which is performed based on a binary string through horizontal flip. In order to facilitate key transfer, we transformed binary string to a number of base 10. Because of the deficiency of Arnold transform in image crop, we used a sign in image center so that we could simulate missing pixels. We measured the transparency of stego image using PSNR index, obtained value was between 40 to 50 dB which was satisfactory. Also, the similarity of secret image and extracted image was measured through NCC and there was more than 99% similarity between them. Then we examined the robustness of steganography to a series of geometric and image processing attacks which yielded good results.

References

[1] R Roy, et al. *Evaluating image steganography techniques: Future research challenges. Computing, Management and Telecommunications (ComManTel).* International Conference on. Ho Chi Minh City, Vietnam. 2013: 309-314.
[2] K Baily, et al. An Evaluation of Image Based Steganography Methods using visual inspection and automated detection techniques. *Multimedia Tools & Applications*. 2006; 31(3): 327-327.

[3] A Kumar, et al. Steganography - A Data Hiding Technique. *International Journal of Computer Applications*. 2010; 9(7): 19-23.
[4] Wikipedia. Wavelet. 2016. *http://en.wikipedia.org/wiki/Wavelet.*
[5] Wikipedia. Haar Wavelet. May 2016. *http://en.wikipedia.org/wiki/Haar_wavelet.*
[6] RC Gonzalez, et al. Digital Image Processing. 3rd Edition. Prentice Hall. 2008.
[7] H Gong, et al. Novel robust blind image watermarking method based on correlation detector. *ICTC.* Seoul. 2011: 751-755.
[8] P Lenarczyk, et al. *Novel Hybrid Blind Digital Image Watermarking in Cepstrum and DCT Domain*. International Conference on Multimedia Information Networking and Security. Jiangsu. 2010: 356-361.
[9] Z Li, et al. *A new blind robust image watermarking scheme in SVD-DCT composite domain.* 18th IEEE International Conference on Image Processing. Brussels. 2011: 2757-2760.
[10] TT Nguyen, et al. *A novel technique for geometrically robust blind image watermarking extraction.* International Conference on Advanced Technologies for Communications (ATC 2013). Ho Chi Minh City. 2013: 101-105.
[11] H Ebrahimi, et al. *Blind Image Watermarking Using Wavelet Coefficient Distance.* 7th Iranian Conference on Machine Vision and Image Processing. Tehran. 2011: 1-5.
[12] KR Kakkirala, et al. *Block based robust blind image watermarking using discrete wavelet transform.* Signal Processing & its Applications, IEEE 10th International Colloquium on. Kuala Lumpur. 2014: 58-61.
[13] TM Thanh, et al. *A proposal of novel q-DWT for blind and robust image watermarking*. IEEE 25th Annual International Symposium on Personal, Indoor, and Mobile Radio Communication (PIMRC). Washington DC. 2014: 2061-2065.
[14] L Yanshan. *A new color image blind watermarking algorithm based on quaternion*. IEEE 10th International Conference on Signal Processing Proceedings. Beijing. 2010: 1698-1701.
[15] Y Zhao, et al. *Multipurpose Blind Watermarking Algorithm for Color Image Based on DWT and DCT.* Wireless Communications, Networking and Mobile Computing (WiCOM), 8th International Conference on. Shanghai. 2012: 1-4.
[16] GS Kalra, et al. *Robust Blind Digital Image Watermarking Using DWT and Dual Encryption Technique*. Computational Intelligence, Communication Systems and Networks (CICSyN), Third International Conference on. Bali. 2011: 225-230.
[17] MIH Sarker, et al. *An improved blind watermarking method in frequency domain for image authentication.* Informatics, Electronics & Vision, International Conference on. Dhaka. 2013: 1-5.
[18] W Yongqi, et al. *A Color Image Blind Watermarking Algorithm Based on Chaotic Scrambling and Integer Wavelet*. Network Computing and Information Security (NCIS), 2011 International Conference on. Guilin. 2011: 413-416.
[19] S Atawneh, et al. Hybrid and Blind Steganography Method for Digital Images Based on DWT and Chaotic Map. *Journal of Communications.* 2013; 8(11): 690-699.
[20] E Chrysochos, et al. *Robust Watermarking of Digital Images Based on Chaotic Mapping and DCT*. Signal Processing Conference, 2008 16th European. Lausanne. 2008: 1-5.
[21] Z Peng, et al. Color Image Authentication Based on Spatiotemporal Chaos and SVD. *Chaos, Solitons & Fractals*. 2008; 36)4): 946-952.
[22] Wikipedia. Arnold's Cat Map. May 2016. *http://en.wikipedia.org/wiki/Arnold%27s_cat_map.*