Image Steganography System Using Bezier Curve

Abdulameer A. Karim[*] ,Ph.D (Asst.Prof). **Abdul Mohsin J. Abdul [*]Hussein[*], Ph.D (Asst.Prof.)**

Haider Mohammed Alwan[*]

Abstract: Securely transmitting data through the internet is performed with the use of steganography the idea of digital image's information sharing. It can be defined as the science and art of disguising data in an ordinary cover media in a manner which doesn't raises doubts of an observer. In this paper the suggested system is modeled for hiding an image in multi-image based on Bezier Curve, this approach makes use of Bezier curve equation in order to select secret image pixel locations and hide it into N cover pixels image. The proposed hiding system gives higher security and can hide larger image into multi small images. The quality of the stego images after data hiding is evaluated using PSNR and MSE and results show that this method produces a precise stego images.

Keywords: Image Steganography, LSB, Image secret, Cover Image, Bezier Curve, Embedding Process, extraction process.

[*] Computer Science Department, University of Technology

1. Introduction

In the beginning of computer invention (the 40's), data security wasn't a subject of importance. Computers were not connected in the form of a net-work, and for stealing data from the computer, it was inevitable entering the computer room, and the security was focused on the building instead of the data or the computer[1]. The invention of the Internet is considered as one of the biggest innovations in the modern ages, information became available online, any user that has a computer is capable of easily connecting to the Internet and start searching for the information he or she wishes to find. Hence, in the latest years there was a continuously raising interest to the means of hiding information within other information, therefore motivated towards studying and finding ways for secret communication[2]. Information hiding is the study of hiding within a cover. This cover (often digital media) may be a text, image, audio, video, Internet packets … etc. By using these covers, communication will be invisible, this part of information hiding is called steganography, although steganography is a very old craft, the honest of computer technology has given it new life. Although steganography is considered as akin to cryptography, steganography may use cryptography techniques to encrypt data before hiding them[1]. The present image steganography techniques embed single image hiding into single cover, as the secret image is directly inserted into the cover media without any additional camouflage except key. The novel idea proposed here is using multi cover, the new multi cover technique employs secret image insertions into multi cover images. In which the path of Bezier curve is used to select the location of the secret image, which will be hidden in to multi cover images. Using multi cover images instead of single cover will increase the payload of the cover images, beside it will increase the complexly of the third party to recover the hidden image, since obtaining one, two or three cover image will not enable him to obtain the secret image.

2. Steganography

Steganography is a kind of secure communication which precisely translates as "covered writing" (it derives its origins from the word stegano or "covered" and the word graphs or "to write" which are of the Greek origin). The aim of steganography is to hide a message in an ordinary cover file such that it's not possible even detecting the existence of that

secret message[3]. Very often along the history, encrypted messages were intercepted but they haven't been deciphered. While this gives a protection to the information that is hidden within the cipher, intercepting the message might be equally harmful due to the fact that it gives an enemy the information that someone is having communication with someone else. Steganography chooses the other method attempting the hiding of any clues that communication is happening[4]. Basically, the hiding procedure in a stenographic system begins with the identification of a cover medium's repeated bits (the ones which might be altered with no tempering with the integrity of that medium). The procedure of hiding generates a steganographic medium via replacing those repeating bits with data from the message that needs to be hidden. The aim of modern steganography is keeping its mere existence unnoticeable, but stenographic systems, due to their invasive nature, leave behind noticeable clues in the cover medium via the modification of its statistical features, therefore, intruders are capable of detecting the distortions in the resulted steganographic medium's statistic features. The procedure of figuring out those distortions is known as Statistical Steganalysis[5]. When a steganographic system is developed, it's necessary considering what the most proper cover Work has to be, in addition to the way the stego-gramme is to reach the receiver. Concerning the development, Steganography is made up of two algorithms, the first one for embedding and the second one is for the extraction. The embedding procedure is involved with the process of embedding a private data inside a cover, and is the most carefully generated procedure of both[6]. A Steganographic algorithm includes a combination of the cover message with the inserted one, which is a thing to be embedded in the cover medium. The algorithm may, or may not, utilize a Steganographic key (stego key) that can be considered as an additional private data which could be needed during the procedure of hiding. An identical key (or a related one) is typically needed for the extraction of the inserted message once more. A typical Steganography system is shown in figure (1).

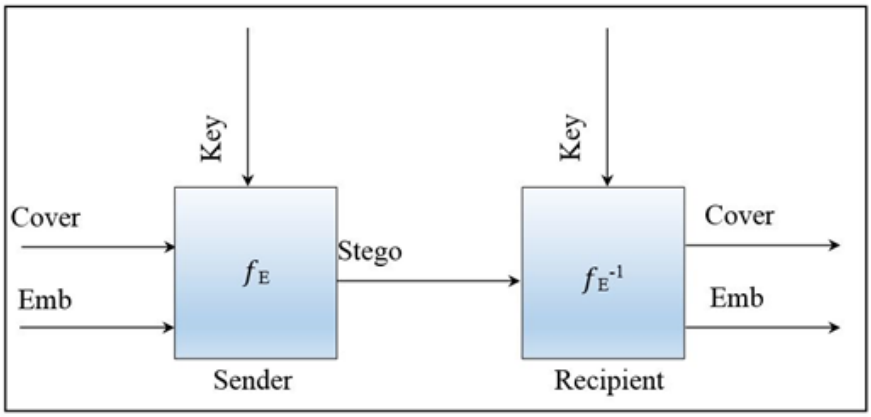


Figure (1): General Steganography System

3. Steganography using LSB

Image Steganography is a growing research area of information security where secret information is embedded in innocent-looking public communication[10]. The image used to camouflage the secret data is called the cover-image[11], while the cover image with which the secret message is transmitted is called as the stego-image[12]. Image or spatial domain techniques embed secret information within the intensity value of the cover image pixels directly[13]. The least significant bit (LSB) substitution is an example of spatial domain techniques. The basic idea in LSB is the direct replacement of LSBs of noisy or unused bits of the cover image with the secret message bits. LSB is the most preferred technique used for data hiding because it is simple to implement offers high hiding capacity,

and provides a very easy way to control stego-image quality, but it has low robustness to modifications made to the stego-image such as low pass filtering and compression and also low imperceptibility[14]. It is the most common and also it is a high capacity steganographic method, but it is not so much robust against certain attacks like low pass filtering and image compression[15]. See figure (2).

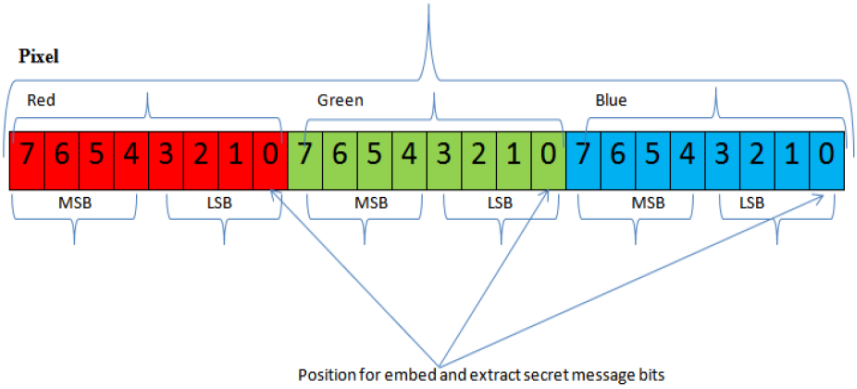


Figure (2): Spatial Operation for LSB

LSB is one of the simplest techniques used for steganography. Images are composed of pixels. The pixel values can be digitally expressed as 0"s and 1"s. For example, RGB 24-bit image has 8 bits representing each of the red, green and blue components. To insert an A (binary value 10000011) into a 24-bit image uses the RGB color model. Each pixel uses eight bits for the intensity of red, green and blue. Three pixels are needed for hiding the letter A. Table (1) and table (2) explain LSB operations in image pixels[16].

Table (1): Before Embedding "A"

	Red Component	Green Component	Blue Component
Pixel 0	00100111	11101001	11001000
Pixel 1	00100111	11001000	11101001
Pixel 2	11001000	00100111	11101001

The changed sequence with the letter A (bit sequence 10000011) embedded would look like this.

Table (2): After Embedding "A"

	Red Component	Green Component	Blue Component
Pixel 0	00100111	11101000	11001000
Pixel 1	00100110	11001000	11101001
Pixel 2	11001001	00100111	11101000

4. Bezier Curve

It's a parametrical curve P(t) which is a polynomial function of the parameter t. The degree of the polynomial depends on the number of points utilized for the identification of the curve[7]. The approach uses control-points and produces an approximation curve. This curve does not pass throughout the local points, instead it's attracted by those points. It's like the points exert a pulling on the curve. Each point influences the direction of the curve by attracting it to itself, and this attracting is at its fullest when the curve becomes as near to the point as possible. Figure (3) shows an illustrate of cubic Bézier curve. This curve is distinguished with 4 points and is a cubic polynomial. Which makes it easier editing, modifying and reshaping the curve, which is one of the reasons why it's popular. The curve might also be modified via the addition of new points, or via eliminating points [8, 9].

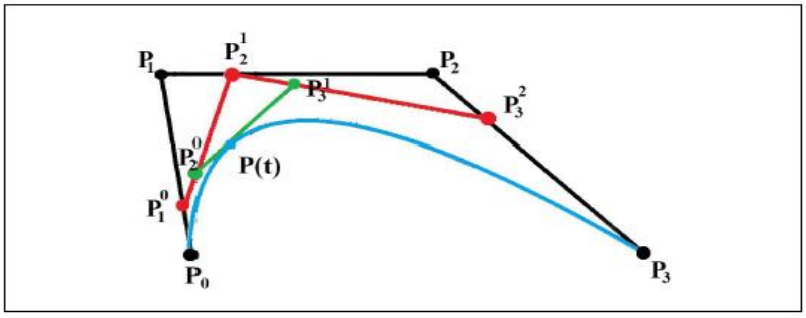


Figure (3): Bezier Curve

The parametrical formula for the cubic Bézier curve using four control points is below:

$$P(t) = (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t)P_2 + t^3 P_3 \quad (1)$$

$$p(x,y) = \begin{pmatrix} fx(t) \\ fy(t) \end{pmatrix}$$

Where ($0 \leq t \leq 1$)

$$fx\,(t) = t3\ x0 + 3\,(1-t)\ t2\ x1 + 3\,(1-t)\ 2\ t\ x2 + (1-t)\ 3x3 \quad (2)$$

$$fy\,(t) = t3\ y0 + 3\,(1-t)\ t2\ y1 + 3\,(1-t)\ 2\ t\ y2 + (1-t)\ 3y3 \quad (3)$$

5. Evaluation Measurements

The calculation of the imperceptibility of a steganographic system two measurements are utilized. Those measurements indicate the degree of similarity (or difference) of the cover image to stego-Image[12]. These measurements are:

- Mean Square Error (MES): The MSE is calculated by comparing byte by byte of the cover image and stego-image. This calculation can be represented by the following equation:

$$MSE\,(f, g) = \frac{1}{MN}\left(\sum_{i=0}^{M}\sum_{j=0}^{N}((f_{ij} - g_{ij})^2\right) \quad (5)$$

  Where M, N denote the number of rows and columns in the original matrix respectively, f_{ij} is the pixel value from cover Image, and g_{ij} is the pixel value from the stego-image. The bigger the value of mean square error the bigger is the dissimilarity between compared images.

- Peak Signal-to-Noise Ratio (PSNR): is used for the calculation of the quality between original image and stego-image. The higher the value of the Peak Signal-to-Noise Ratio (PSNR) is, the better is the quality. PSNR is calculated with the use of the equation

$$PSNR = 10.\,Log10\left(\frac{255^2}{MSE\,(f,g)}\right) \quad (4)$$

6 Proposed System

In the traditional LSB the secret image pixels will be embedded sequentially i.e. the first pixel of the secret image will be embedded in the first pixel of the cover image and soon, thus the third party can easily find out which pixels of the secret image is stored in the cover image. The below steps illustrate the traditional LSB embedding as show figure (4):

1- Load covers image and secret image.
2- Convert secret image from color image to gray image.
3- Read contents the first location pixel from cover image and secret image.
4- Suppose the first location pixel of cover image contents Red =115, Blue=210, Green = 80 and Alpha = 255.
5- Suppose the first location pixel of secret image contents is equal 35.
6- Converting all value in step (4, 5) to binary value and embedding secret image value in to cover image.
   115 = 0 1 1 1 0 0 1 1, 210 = 1 1 0 1 0 0 1 0, 80 = 0 1 0 1 0 0 0 0, 255 = 1 1 1 1 1 1 1 1, 110 = 0 1 1 0 1 1 1 0.
7- Embedding = 0 1 1 0 1 1 1 0 in to cover image the value after embedding is Red = 0 1 1 1 0 0 0 1, Blue = 1 1 0 1 0 0 1 0, Green = 0 1 0 1 0 0 1 1 and Alpha = 1 1 1 1 1 1 1 0.
8- Converting all valve in step (8) to decimal value thus Red = 113, Blue = 210, Green = 83 and Alpha = 254.
9- After applying the steps (7-9) to all the image the output will be stego_ image.

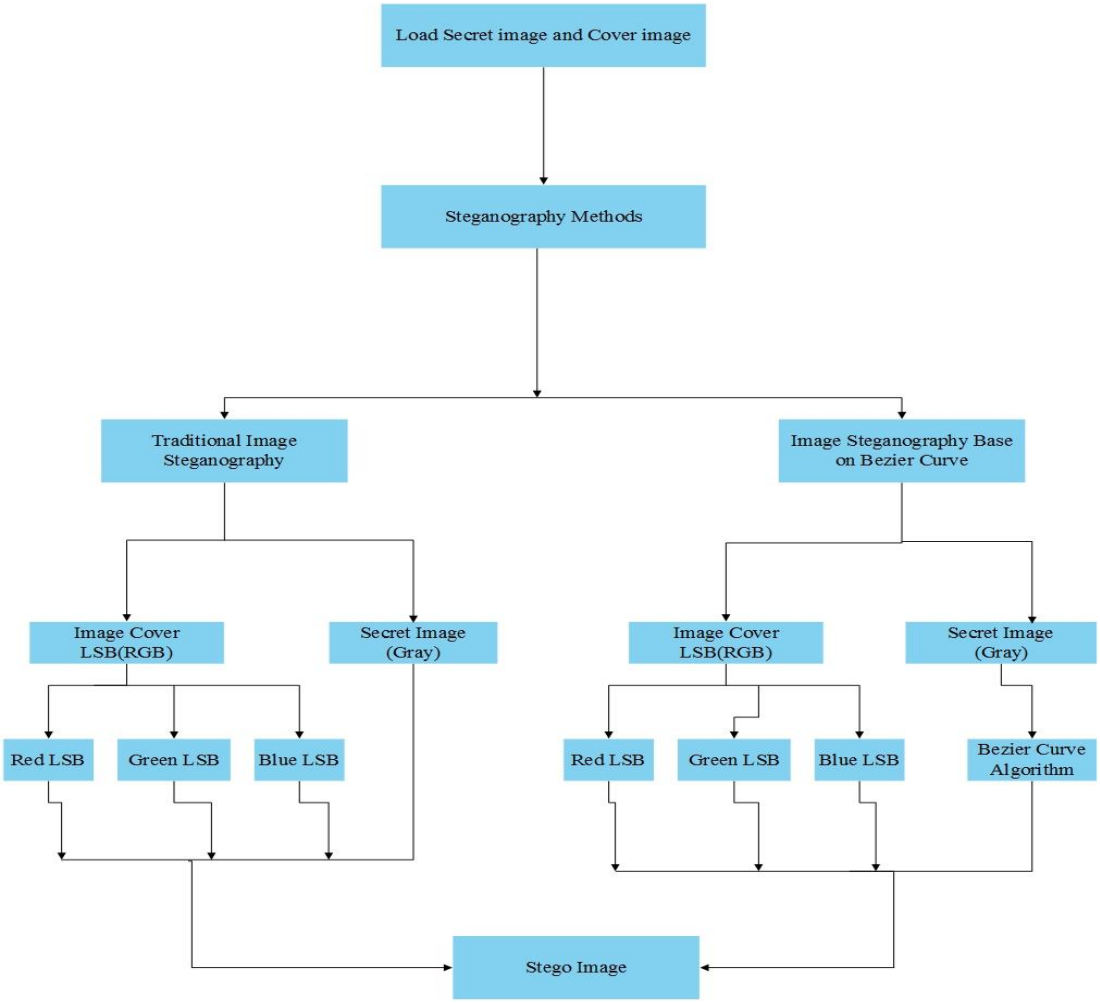
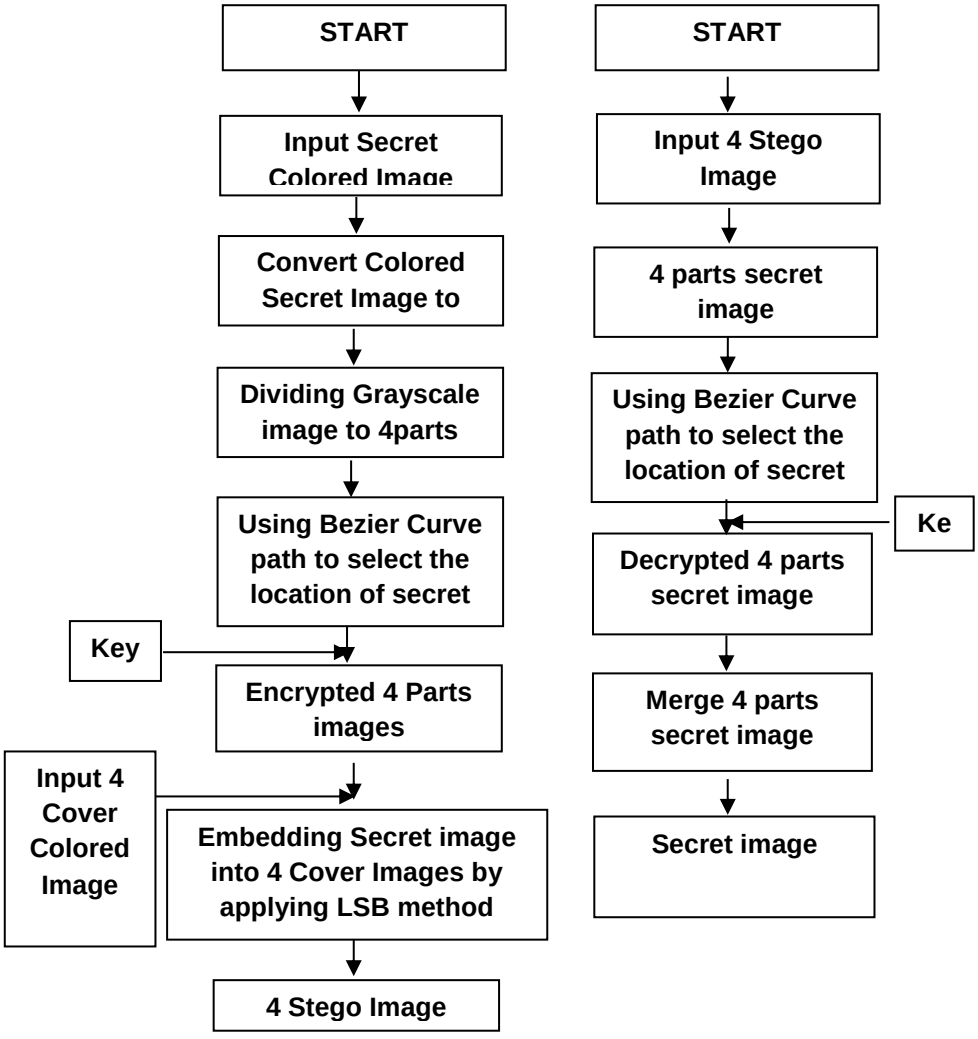


Figure (4): General Scheme of the System

The designed and executed system does two main functions, Embedding and Extraction. These two functions based on several sub functions in order to complete the job, which can be shown as figure (5)



a. Hiding Image b. Recover Image

Figure (5): General System Block Diagram

6.1 Hiding Image (Steganography) Using Bezier Curve

This is the main procedure in the system which has been implemented using the following steps:

1. Input four color cover images and grayscale secret image and key.
2. Define Some Variables such as Img_H, P_C_Img.
3. Select location of the pixel from secret image based on output X, Y after applying equation (2) and equation (3).
4. Converting content of this location to bits, executing XOR logical operation with Key by applying convert Byte to bits operation, and put result in Message_bits array.
5. Converting content of alpha, red, green and blue to bits and executing XOR logical operation with Key by applying convert Byte to bits operation put the result in Alpha_ bits array, Red_bits array, Green_bits array and Blue_bits array.
6. Converting content of alpha, red, green and blue to bits and executing XOR logical operation with Key by applying convert Byte to bits operation put the result in Alpha_ bits array, Red_bits array, Green_bits array and Blue_bits array.
7. After converting content of pixel location, alpha, red, green and blue to bits by applying Byte to bits operation, put all result in Message_bits array, Alpha_bits array, Red_bits array, Green_bits array, Blue_bits array, Replace the value of location 6 and 7 of arrays Alpha_bits, Red_bits, Green_bits and Blue_bits with the value of the location 0 to 7 of the Message_bits array, so every bit of the secret image will be hidden.
8. Convert bits for Alpha_bits array, Red_bits array, Green_bits array and Blue_bits array after hiding Message_bits array to bytes, logical operation XOR will be performed on this byte and the key by applying convert bits to Byte operation and output from this algorithm is draw stego image.

All these steps can be implemented by using the algorithm 1 below:

Algorithm1: Hide Image (Steganography) Using Bezier Curve

Input: Cover_Image (C_Img), Secret_Image (S_Img), Key (K)
Output: Stego_Imge (Ste_Img).

```
Image_Hide (Img_H), PixelContainerImage (P_C_Img),
PixelMsgImage (P_M_Img)
Begin
Step1: Define Variables
        Defines: K = 0,
         Img _H = new image (S_Img.Width, S_Img.Height),
        P_C_Img = new Color, P_M_Img = new Color.
        byte [ ] Msgbits, byte[ ] Alphabits, byte[ ] Redbits,
        byte[ ] Greenbits, byte[ ] Bluebits
        byte newAlpha = 0,byte newRed = 0, byte newGreen = 0,
        byte newBlue = 0;
Step2: Secret Image Embedding
        For i = 0 to S_Img.Width.
        For j = 0 to S_Img.Height.
        Apply Draw Bezier curve operation.
        IF X >= S_Img.Width.
        X = i.
        End IF.
        IF Y >= S_Img.Height.
        Y = j
        End IF.
        P_M_Img = S_Img.getPixel (X, Y).
      Msgbits = result of applying Convert Byte to bits operation
      [Byte (P_M_Img.R), K].
        P_C_Img = C_Img.getpixel (i, j).
    Alphabits = result of applying Convert Byte to bits operation
        [Byte (P_C_Img.A), K].
    Redbits = result of applying Convert Byte to bits operation
        [Byte (P_C_Img.R), K].
    Bluebits = result of applying Convert Byte to bits operation
        [Byte (P_C_Img.B), K].
  Greenbits = result of applying Convert Byte to bits operation
        [Byte (P_C_Img.G), K].
        Alphabits[6] = Msgbits[0];
          Alphabits[7] = Msgbits[1];
        Redbits[6] = Msgbits[2];
```

```
            RedBits[7] = Msgbits[3];
            Greenbits[6] = MsgBits[4];
            Greenbits[7] = Msgbits[5];
            Bluebits[6] = Msgbits[6];
            Bluebits[7] = Msgbits[7];
        newAlpha = result of applying Convert bits to Byte operation
            [Alphabits, K].
        newRed = result of applying Convert bits to Byte operation
            [Redbits, K].
        newGreen = result of applying Convert bits to Byte operation
            [Greenbits, K].
        newBlue = result of applying Convert bits to Byte operation
            [Bluebits, K].
            Draw P_C_Img = (newAlpha, newRed, newGreen,
            newBlue).
            Steg_Img. SetPixel (i, j, P_C_Img).
            End
```

6.2 Recovering Image (Extraction) Using Bezier Curve

This is inverse of the main procedure of the system, in which the input stego image and the output is four parts of images, these parts represent the secret image extracted from the covers inserted by applied algorithm hiding image, After extracting four parts of the image will be applied algorithm merge to merge all these parts to obtain the secret image.

All these steps can be implemented by using the algorithm 2 below:

Algorithm 2: Recover Image (Extraction) Using Bezier Curve

```
Input: Stego_Imge (Ste_Img), Key (K)
Output: Secret_Image (S_Img).
```

```
Pixel_Decrypt (P_D)
Begin
Step1: Define Variables
        Defines: K = 0,
         S_Img = new image (Ste_Img.Width, Ste_Img.Height),
        P_D= new Color,
        byte [ ] P_D , byte[ ] AlphaBits, byte[ ] RedBits,
        byte [ ] GreenBits, byte[ ] BlueBits, byte NewGray = 0.
```

```
Step2: Image Recover
        For i = 0 to Ste_Img.Width.
        For j = 0 to Ste_Img.Height.
        Apply Algorithm (Draw Bezier Curve).
        IF X >= Ste_Img.Width.
        X = i.
        End IF.
        IF Y >= Ste_Img.Height.
        Y = j
        End IF.
        P_D = Ste_Img.getPixel (i, j).
      Alphabits = result of applying Convert Byte to bits operation
                        [Byte (P_D.A), K].
        Redbits = result of applying Convert Byte to bits operation
                        [Byte (P_D.R), K].
        Bluebits = result of applying Convert Byte to bits operation
                        [Byte (P_D.B), K].
      Greenbits = result of applying Convert Byte to bits operation
                        [Byte (P_D.G), K].
        P_D [0] = Alphabits [6].
        P_D [1] = Alphabits [7].
        P_D [2] = Redbits [6].
        P_D [3] = Redbits [7];
        P_D [4] = Greenbits [6].
        P_D [5] = Greenbits [7].
        P_D [6] = Bluebits [6].
        P_D [7] = Bluebits [7].
        NewGray = result of applying Convert bits to Byte operation
        [P_D, K].
        Draw P_D = (NewGray, NewGray, NewGray).
        S_Img. SetPixel (X, Y, P_D).
End
```

7 Results and Discussions

The secret image is divided into N parts equal to number cover images and use LSB method to hide it. Bezier Curve equation has been used in order to select secret image pixel locations and hide it into of N cover images pixels. Several tests are applied to evaluate the performance of the proposed system. For this reason a dataset consists of images represent secret image (10 images different in sizes and different

file format type) has been built. Some images of this dataset are standard such as Lena, Pepper while the rest of the images are selected in a random manner according to various properties figure (6) illustrates the secret images dataset. Table (3) shows the properties of images, In addition, the data-set of various images represent cover image (40 images different in sizes and file format type) has been built. Figure (7) illustrates the cover images dataset. Table (4) shows the properties of images.

Figure (6): Secret image Dataset

Table (3): The Characteristics of the Secret Images

Image Number	Image Size	File Format
1	512*512 (462KB)	PNG
2	800*600 (111KB)	PNG
3	1024*720 (112KB)	PNG
4	800*600 (1.37MB)	BMP
5	800*600 (73.4KB)	BMP
6	800*600 (1.37MB)	BMP
7	800*600 (1.37MB)	BMP
8	512*512 (47.3)	JPEG
9	1920*1080 (143KB)	JPEG
10	1920*1080 (1.08MB)	JPEG

Figure (7): Cover Images Dataset

Table (4): Cover Images Characteristics

Image Number	Image Size	File Format
1	500*375 (345KB)	BMP
2	680*591 (24KB)	BMP
3	500*375 (31KB)	BMP
4	400*300 (46KB)	BMP
5	800*575 (115KB)	BMP
6	400*300 (65KB)	BMP
7	400*300 (115KB)	BMP
8	600*337 (45KB)	BMP
9	400*300 (115KB)	BMP
10	603*390 (101KB)	BMP
11	256*256 (12KB)	JPEG
12	960*600 (143KB)	JPEG
13	960*600 (186KB)	JPEG
14	960*600 (65KB)	JPEG
15	960*600 (165KB)	JPEG
16	256*256 (12KB)	JPEG
17	256*256 (15KB)	JPEG
18	256*256 (16KB)	JPEG
19	960*540 (157KB)	JPEG
20	960*540 (158K)	JPEG
21	960*540 (144KB)	JPEG
22	960*540 (74KB)	JPEG
23	960*540 (108KB)	JPEG
24	960*540 (38KB)	JPEG
25	256*256 (15KB)	PNG
26	256*256 (17KB)	PNG
27	256*256 (10KB)	PNG
28	256*256 (14KB)	PNG
29	400*300 (67KB)	PNG
30	400*300 (11KB)	PNG
31	400*300 (126KB)	PNG
32	400*300 (27KB)	PNG
33	512*360 (417KB)	PNG
34	360*512 (194KB)	PNG
35	512*360 (403KB)	PNG
36	512*360 (249KB)	PNG
37	960*540 (210KB)	PNG

38	960*540 (333KB)	PNG
39	960*540 (123KB)	PNG
40	960*540 (123KB)	PNG

When the system is run the first part of the implementation phase, will be performed in which main form of proposed system occur then the user choose four cover images with one secret image as input as shown figure (8).

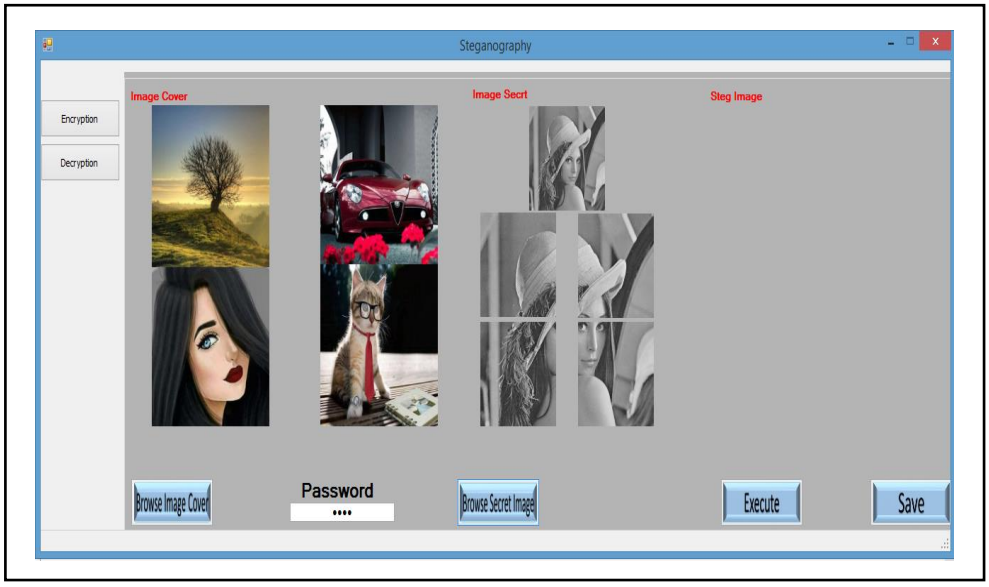


Figure (8): Uploading Cover Images and secret image

After dividing Grayscale secret image to four parts, figure 9 illustrate applying Bezier curve equation into secret image parts depending on the passage of the Bezier curve secret image pixels that are chosen that which gives random selection in pixel location of the secret image parts.

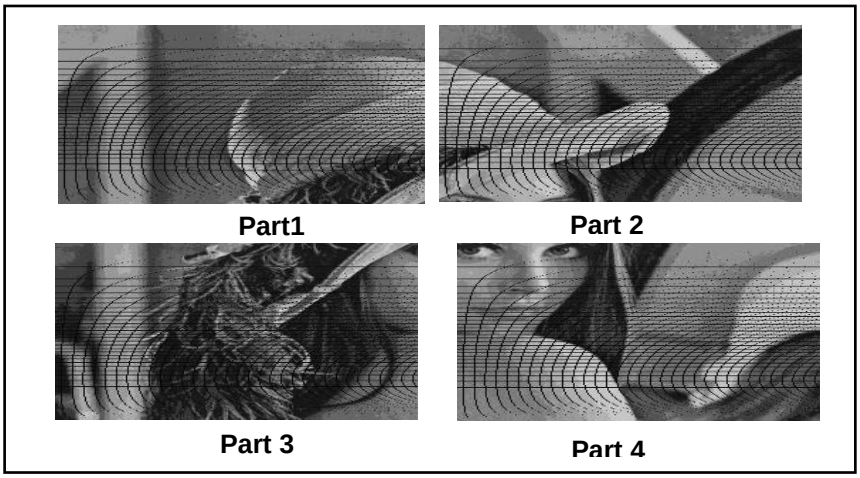


Figure (9): Grayscale secret image divided to 4 parts and applying Bezier curve equation.

After applying algorithm1 (Hide Image) between secret image and four cover images the output are four stego images as illustrate in figure 10**.**

Figure (10): Stego image.

After obtaining the stego image performance evaluating of stego image has been implemented through two measuring PSNR and MSE as show in the table 5.

Table (5): PSNR and MSE

Secret Image	Cover Image	PSNR of cover image to Stego image(DB) using Bezier Curve	MSE of cover image to stego image using Bezier Curve	PSNR of cover image to Stego image(DB) LSB Traditional	MSE of cover image to stego image LSB Traditional
1	Cover1	46.25	1.54	39.12	7.96
	Cover2	46.92	1.32	41.60	4.49
	Cover3	46.30	1.52	41.07	5.08
	Cover4	46.69	1.39	41.12	5.02
2	Cover5	46.98	1.30	40.09	6.36
	Cover6	46.74	1.37	39.14	7.92
	Cover7	46.63	1.41	38.97	8.24
	Cover8	47.01	1.29	40.85	5.34
3	Cover9	47.55	1.14	42.10	4.00
	Cover10	47.10	1.26	40.90	5.28
	Cover11	45.81	1.70	39.79	6.82
	Cover12	46.63	1.41	42.06	4.04
4	Cover13	46.65	1.40	40.92	5.26
	Cover14	46.96	1.30	40.27	6.11
	Cover15	46.45	1.47	39.90	6.65
	Cover16	46.43	1.47	42.30	3.82
5	Cover17	47.45	1.16	39.72	6.93
	Cover18	46.46	1.46	42.12	3.99
	Cover19	45.52	1.82	41.57	4.52
	Cover20	46.28	1.53	40.90	5.28
6	Cover21	46.18	1.56	39.67	39.67
	Cover22	46.64	1.40	41.19	4.94
	Cover23	46.30	1.52	40.57	5.70
	Cover24	46.66	1.40	42.95	3.29
7	Cover25	43.41	2.96	41.51	4.59
	Cover26	46.71	1.38	39.98	6.53
	Cover27	46.78	1.36	43.01	3.25
	Cover28	46.48	1.46	42.38	3.75
8	Cover29	46.18	1.56	42.97	3.28
	Cover30	46.44	1.47	41.24	4.88
	Cover31	47.30	1.21	42.37	3.76

	Cover32	47.61	1.12	42.94	3.30
9	Cover33	46.14	1.58	42.40	3.74
	Cover34	47.12	1.26	41.67	4.42
	Cover35	46.57	1.43	41.68	4.41
	Cover36	45.73	1.73	41.00	5.16
10	Cover37	47.34	1.19	42.09	4.01
	Cover38	47.90	1.05	41.68	4.41
	Cover39	46.33	1.51	39.34	7.56
	Cover40	46.67	1.39	41.21	41.21

8 Conclusion

1) Least Significant Bit (RGB) algorithm provides a good security, and good quality which has been measured by PNSR, MSE.
2) Hiding secret image into multi-cover will increase the hiding security i.e. if the third party discover one or more of the cover images he can't obtain the secret image unless he find the total number of the cover images.
3) Using more than one cover images give the possibility of carrying much more secret data (Payload) than traditional method.
4) Using Bezier Curve algorithm give a high random distribution to the pixels of the secure image when hiding it on the pixels of the cover image.

References

[1] Al- Ethawi G. A. S. ,2002 "Text hiding in image border" Msc, Dean of the Military College of Engineering, November.

[2] Richard P, 1998," An Analysis of Steganographic Techniques ", Msc, Computers Department of Computer Science and Software Engineering. The University of Timisoara Faculty of Automatics.

[3] Lu S., 2005 " Steganography and Digital Watermarking Techniques for Protection of Intellectual Property", Idea Group Publishing.

[4] Popa R., 1998 "An Analysis of Steganographic Techniques," Working Report on Steganography, Faculty of Automatics and Computers, Vol.7, No. 2.

[5] Provos N. and Honeyman P., 2003 “Hide and Seek: An Introduction to Steganography,” Computer Journal of IEEE Security and Privacy Magazine, 2(3), pp: 32-40.

[6] Philip B., 2008" Image Steganography And Steganalysis", Department of Computing Faculty of Engineering and Physical Sciences University of Surrey Guildford Surrey United Kingdom, August.

[7] Samir K. B., Debnath. B., Debashis G., Swarnendu M. and Poulami D., 2008" A Tutorial Review on Steganography", University of Calcutta, UFL & JIITU, IC3, pp: 107-108.

[8] Morkel T., Eloff J.H.P. and Olivier M.S., 2005" An Overview Of Image Steganography", Information and Computer Security Architecture (ICSA) Research Group, Department of Computer Science, University of Pretoria, 0002, Pretoria, South Africa, June/July.

[9] Thomas W. Sederberg, “Computer Aided Geometric Design”, Springer Science Business Media, October 9, 2014.

[10] Muhammad K., Ahmad J., Farman H. And Jan Z.,” A New Image Steganographic Technique using Pattern based Bits Shuffling and Magic LSB for Grayscale Images”, Sindh University Research Journal, Vol. 47, 2015.

[11] Al-Tamimi A. T. and Alqobaty A. A.,” Image Steganography Using Least Significant Bits (LSBs): A Novel Algorithm”, International Journal of Computer Science and Information Security, Vol. 13, No. 1, January 2015.

[12] Swain G., ” Digital Image Steganography using Nine-Pixel Differencing and Modified LSB Substitution”, Indian Journal of Science and Technology, Vol. 7, September 2014.

[13] Jain M. and Lenka S.K.,” A Review of Digital Image Steganography using LSB and LSB Array”, International Journal of Applied Engineering Research, Vol. 11, No. 3, 2016.

[14] Jamdar A.S., Shah A.V., Gavali D.D. and Kurkute S.L.,” Edge Adaptive Steganography Using DWT”, International Journal of Engineering and Advanced Technology, Vol.-2, Issue 4, April 2013.

[15] Mazumder J. A., Hemachandran K.,” Color Image Steganography Using Discrete Wavelet Transformation and Optimized Message Distribution Method”, International Journal of Computer Sciences and Engineering, Vol. 7, July 2014.

[16] Yang X., “Review of Metaheuristics and Generalized Evolutionary Walk Algorithm”, Int. J. Bio-Inspired Computation, Vol. 3, No. 2, 2011.

نظام الاخفاء الصور باستخدام منحي البيزاييز

أ.م. د.عبدالأمير عبدالله كريم* أ.م.د. عبد المحسن جابر عبد الحسين*

الباحث: حيدر محمد علوان*

المستخلص: يتم نقل البيانات بشكل آمن من خلال شبكة الإنترنت باستخدام فكرة ستيغانوغرافي (Steganography) لمشاركة معلومات الصورة الرقمية. ويمكن تعريف هذه الفكرة على أنها علم وفن تمويه البيانات في وسائط مرئية مثلا الصور او الفيديو او الصوت بطريقة لا تثير شكوك المراقب. في هذه البحث تم تصميم النظام المقترح لإخفاء صورة في صور متعددة على أساس منحنى بيزاييز (Bezier) Curve، وهذا النهج يجعل من استخدام معادلة منحنى بيزاييز من أجل تحديد مواقع بكسل الصورة السرية وإخفاءها في N من اغطية الصور. نظام الاخفاء المقترح يعطي امنية عالية ويمكن إخفاء صورة ذات حجم كبير في مجموعة صورة صغيرة. يتم تقييم جودة الصور Stego بعد إخفاء البيانات باستخدام معايير التقييم وهي PSNR وMSE وتبين النتائج ان هذه الطريقة أكثر دقة.

الكلمات المفتاحية: صورة إخفاء المعلومات، LSB، صورة السر، صورة الغلاف، منحنى بيزير، عملية التضمين، عملية الاستخراج.

*قسم علوم الحاسوب / الجامعه التكنولوجيه