International Journal of Scientific Engineering and Technology (ISSN : 2277-1581)
www.ijset.com, Volume No.1, Issue No.4, pg : 127-133 01 Oct. 2012

SECURING THE INFORMATION USING HYPER-CHAOS ENCRYPTION ALGORITHM

[1]K.Mohan, [2]S.Lakshmi, [3]M.Sumithra, [4]S.Muruganandam

[1, 3, 4]Department of Computer Science & Engg., [2]Department of Electronics and Comm. Engg.

[1, 2, 4] Thirumalai Engineering College, Kanchipuram

[3]Jei Mathaajee College of Engineering, Kanchipuram

[1]jackmoh2000.2009@gmail.com, [2]msklakshmi@gmail.com, [3]sumihaifriends@gmail.com , [4]murugansoft@hotmail.com

ABSTRACT

Steganography is the art of hiding the fact that communication is taking place, by hiding information in other information. Many different carrier file formats can be used, but digital images are the most popular because of their frequency on the Internet. For hiding secret information in images, there exists a large variety of steganographic techniques some are more complex than others and all of them have respective strong and weak points. Different applications have different requirements of the steganography technique used. In this paper presents hiding the data behind the image then encrypt the image by using hyper-chaos encryption algorithm. In this algorithm, shuffling matrix and diffusing matrix are generated based on Chen's hyper-chaotic system. Firstly, the Chen's hyper-chaotic system is used to shuffle the position of the image pixels, and then use Chen's hyper-chaotic system to confuse the relationship between the original image and the cipher image.

Keywords: Hyper-chaos, Steganography, hiding data

I. INTRODUCTION

Since the rise of the Internet one of the most important factors of information technology and communication has been the security of information. Cryptography was created as a technique for securing the secrecy of communication and many different methods have been developed to encrypt and decrypt data in order to keep the message secret. Unfortunately it is sometimes not enough to keep the contents of a message secret, it may also be necessary to keep the existence of the message secret. The technique used to implement this, is called steganography. Steganography is the art and science of invisible communication. This is accomplished through hiding information in other information, thus hiding the existence of the communicated information. The word steganography is derived from the Greek words "*stegos*" meaning "cover" and "*grafia*" meaning "writing" [1] defining it as "covered writing". In image steganography the information is hidden exclusively in images.

Steganography differs from cryptography in the sense that where cryptography focuses on keeping the contents of a message secret, steganography focuses on keeping the existence of a message secret [4]. Steganography and cryptography are both ways to protect information from unwanted parties but neither technology alone is perfect and can be compromised. Once the presence of hidden information is revealed or even suspected, the purpose of steganography is partly defeated [4]. The strength of steganography can thus be amplified by combining it with cryptography.

Research in steganography has mainly been driven by a lack of strength in cryptographic systems. Many governments have created laws to either limit the strength of a cryptographic system or to prohibit it altogether [7], forcing people to study other methods of secure information transfer. Businesses have also started to realize the potential of steganography in communicating trade secrets or new product information. Avoiding communication through well-known channels greatly reduces the risk of information being leaked in transit [8]. Hiding information in a photograph of the company picnic is less suspicious than communicating an encrypted file.

2. IMAGE STEGANOGRAPHY

As stated earlier, images are the most popular cover objects used for steganography. In the domain of digital images many different image file formats exist, most of them for specific applications. For these different image file formats, different steganographic algorithms exist.

2.1 IMAGE DEFINITION

To a computer, an image is a collection of numbers
that constitute different light intensities in different areas of the image [9]. This numeric representation forms a grid and the individual points are referred to as pixels. Most images on the Internet consists of a rectangular map of the image's pixels (represented as bits) where each pixel is located and its colour [10]. These pixels are displayed horizontally row by row.

The number of bits in a colour scheme, called the bit depth, refers to the number of bits used for each pixel [11].The smallest bit depth in current colour schemes is 8, meaning that there are 8 bits used to describe the colour of each pixel [11]. Monochrome and grayscale images use 8 bits for each pixel and are able to display 256 different colours or shades of grey. Digital colour images are typically stored in 24-bit files and use the RGB colour model, also known as true colour [11]. All colour variations for the pixels of a 24-bit image are derived from three primary colours: red, green and blue, and each primary colour is represented by 8 bits [9]. Thus in one given pixel, there can be 256 different quantities of red, green and blue, adding up to more than 16-million combinations, resulting in more than 16-million colours [11]. Not surprisingly the larger amount of colours that can be displayed, the larger the file size [10].

2.2 IMAGE AND TRANSFORM DOMAIN

Image steganography techniques can be divided into two groups: those in the Image Domain and those in the Transform Domain [2]. Image – also known as spatial – domain techniques embed messages in the intensity of the pixels directly, while for transform – also known as frequency – domain, images are first transformed and then the message is embedded in the image.

Image domain techniques encompass bit-wise methods that apply bit insertion and noise manipulation and are sometimes characterised as “simple systems” [12]. The image formats that are most suitable for image domain steganography are lossless and the techniques are typically dependent on the image format [13]. Steganography in the transform domain involves the manipulation of algorithms and image transforms [12].These methods hide messages in more significant areas of the cover image, making it more robust [4]. Many transform domain methods are independent of the image format and the embedded message may survive conversion between lossy and lossless compression [13]. In the next sections steganographic algorithms will be explained in categories according to image file formats and the domain in which they are performed.

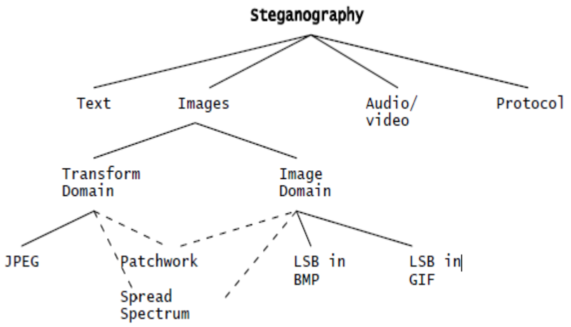

Fig.1.1 Categories of image stenography

2.2.1 Image Domain

- **Least Significant Bit**
- **LSB and Palette Based Images**

- **Least Significant Bit**

Least significant bit (LSB) insertion is a common, simple approach to embedding information in a cover image [12]. The least significant bit (in other words, the 8th bit) of some or all of the bytes inside an image is changed to a bit of the secret message. When using a 24-bit image, a bit of each of the red, green and blue colour components can be used, since they are each represented by a byte. In other words, one can store 3 bits in each pixel. An 800 × 600 pixel image, can thus store a total amount of 1,440,000 bits or 180,000 bytes of embedded data [19]. For example a grid for 3 pixels of a 24-bit image can be as follows:

```
(00101101 00011100 11011100)
(10100110 11000100 00001100)
(11010010 10101101 01100011)
```

When the number 200, which binary representation is 11001000, is embedded into the least significant bits of this part of the image, the resulting grid is as follows:

```
(00101101 00011101 11011100)
(10100110 11000101 00001100)
(11010010 10101100 01100011)
```

Although the number was embedded into the first 8 bytes of the grid, only the 3 underlined bits needed to be changed according to the embedded message. On average, only half of the bits in an image will need to be modified to hide a secret message using the maximum cover size. Since there are 256 possible intensities of each primary colour, changing the LSB of a pixel results in small changes in the intensity of the colours. These changes cannot be perceived by the human eye - thus the message is successfully hidden. With a well-chosen image, one can even hide the message in the least as well as second to least significant bit and still not see the difference.

In the above example, consecutive bytes of the image data – from the first byte to the end of the message – are used to embed the information. This approach is very easy to detect [4]. A slightly more secure system is for the sender and receiver to share a secret key that specifies only certain pixels to be changed. Should an adversary suspect that LSB steganography has been used, he has no way of knowing which pixels to target without the secret key [5].

In its simplest form, LSB makes use of BMP images, since they use lossless compression. Unfortunately to be able to hide a secret message inside a BMP file, one would require a very large cover image. Nowadays, BMP images of 800 × 600 pixels are not often used on the Internet and might arouse suspicion. For this reason, LSB steganography has also been developed for use with other image file formats.

- **LSB and Palette Based Images**

Palette based images, for example GIF images, are another popular image file format commonly used on the Internet. By definition a GIF image cannot have a bit depth greater than 8, thus the maximum number of colours that a GIF can store is 256. GIF images are indexed images where the colours used in the image are stored in a palette, sometimes referred to as a colour lookup table. Each pixel is represented as a single byte and the pixel data is an index to the colour palette. The colours of the palette are typically ordered from the most used colour to the least used colours to reduce lookup time [16].

GIF images can also be used for LSB steganography, although extra care should be taken. The problem with the palette approach used with GIF images is that should one change the least significant bit of a pixel, it can result in a completely different colour since the index to the colour palette is changed. If adjacent palette entries are similar, there might be little or no noticeable change, but should the adjacent palette entries be very dissimilar, the change would be evident [16]. One possible solution is to sort the palette so that the colour differences between consecutive colours are minimized [10]. Another solution is to add new colours which are visually similar to the existing colours in the palette. This requires the original image to have less unique colours than the maximum number of colours (this value depends on the bit depth used) [1]. Using this approach, one should thus carefully choose the right cover image. Unfortunately any tampering with the palette of an indexed image leaves a very clear signature, making it easier to detect. A final solution to the problem is to use greyscale images. In an 8-bit greyscale GIF image, there are 256 different shades of grey. The changes between the colours are very gradual, making it harder to detect.

LSB and Palette Based Images Palette based images, for example GIF images, are another popular image file format commonly used on the Internet. By definition a GIF image cannot have a bit depth greater than 8, thus the maximum number of colours that a GIF can store is 256. GIF images are indexed images where the colours used in the image are stored in a palette, sometimes referred to as a colour lookup table. Each pixel is represented as a single byte and the pixel data is an index to the colour palette. The colours of the palette are typically ordered from the most used colour to the least used colours to reduce lookup time [16]. GIF images can also be used for LSB steganography, although extra care should be taken. The problem with the palette approach used with GIF images is that should one change the least significant bit of a pixel, it can result in a completely different colour since the index to the colour palette is changed [16]. If adjacent palette entries are similar, there might be little or no noticeable change, but should the adjacent palette entries be very dissimilar, the change would be evident [16]. One possible solution is to sort the palette so that the colour differences between consecutive colours are minimized [10]. Another solution is to add new colours which are visually similar to the existing colours in the palette. This requires the original image to have less unique colours than the maximum number of colours (this value depends on the bit depth used) [1]. Using this approach, one should thus carefully choose the right cover image. Unfortunately any tampering with the palette of an indexed image leaves a very clear signature, making it easier to detect.

A final solution to the problem is to use greyscale images. In an 8-bit greyscale GIF image, there are 256 different shades of grey. The changes between the colours are very gradual, making it harder to detect.

2.2.2 Transform Domain

To understand the steganography algorithms that can be used when embedding data in the transform domain, one must first explain the type of file format connected with this domain. The JPEG file format is the most popular image file format on the Internet, because of the small size of the images.

- **JPEG compression**

To compress an image into JPEG format, the RGB colour representation is first converted to a YUV representation. In this representation the Y component corresponds to the luminance (or brightness) and the U and V components stand for chrominance (or colour) [1]. According to research the human eye is more sensitive to changes in the brightness (luminance) of a pixel than to changes in its colour [11]. This fact is exploited by the JPEG compression by down sampling the colour data to reduce the size of the file. The colour components (U and V) are halved in horizontal and vertical directions, thus decreasing the file size by a factor of 2 [1].

The next step is the actual transformation of the image. For JPEG, the Discrete Cosine Transform (DCT) is used, but similar transforms are for example the Discrete Fourier Transform (DFT). These mathematical transforms convert the pixels in such a way as to give the effect of “spreading” the location of the pixel values over part of the image [11]. The DCT transforms a signal from an image representation into a frequency representation, by grouping the pixels into 8 × 8 pixel blocks and transforming the pixel blocks into 64 DCT coefficients each. A modification of a single DCT coefficient will affect all 64 image pixels in that block.

The next step is the quantization phase of the compression. Here another biological property of the human eye is exploited: The human eye is fairly good at spotting small differences in brightness over a relatively large area, but not so good as to distinguish between different strengths in high frequency brightness [1]. This means that the strength of higher frequencies can be diminished, without changing the appearance of the image. JPEG does this by dividing all the values in a block by a quantization coefficient. The results are rounded to integer values and the coefficients are encoded using Huffman coding to further reduce the size [11].

2.3 IMPLEMENTATION OF DOMAINS

As seen in Figure 1, some steganographic algorithms can either be categorised as being in the image domain or in the transform domain depending on the implementation.

- **Patchwork**

Patchwork is a statistical technique that uses redundant pattern encoding to embed a message in an image. The algorithm adds redundancy to the hidden information and then scatters it throughout the image. A pseudorandom generator is used to select two areas of the image (or patches), patch A and patch B. All the pixels in patch A is lightened while the pixels in patch B is darkened. In other words the intensities of the pixels in the one patch are increased by a constant value, while the pixels of the other patch are decreased with the same constant value [6]. The contrast changes in this patch subset encodes one bit and the changes are typically small and imperceptible, while not changing the average luminosity. A disadvantage of the patchwork approach is that only one bit is embedded. One can embed more bits by first dividing the image into sub-images and applying the embedding to each of them. The advantage of using this technique is that the secret message is distributed over the entire image, so should one patch be destroyed, the others may still survive [16]. This however, depends on the message size, since the message can only be repeated throughout the image if it is small enough. If the message is too big, it can only be embedded once.The patchwork approach is used independent of the host image and proves to be quite robust as the hidden message can survive conversion between lossy and lossless compression.

- **Spread Spectrum**

In spread spectrum techniques, hidden data is spread throughout the cover-image making it harder to detect [4]. A system proposed by Marvel et al. combines spread spectrum communication, error control coding and image processing to hide information in images [6]. Spread spectrum communication can be defined as the process of spreading the bandwidth of a narrowband signal across a wide band of frequencies [6]. This can be accomplished by adjusting the narrowband waveform
with a wideband waveform, such as white noise. After spreading, the energy of the narrowband signal in any one frequency band is low and therefore difficult to detect [6]. In spread spectrum image steganography the message is embedded in noise and then combined with the cover image to produce the stego image. Since the power of the embedded signal is much lower than the power of the cover image, the embedded image is not perceptible to the human eye or by computer analysis without access to the original image [6].

3. PROPOSED SYSTEM

3.1 A HYPER-CHAOTIC CHEN'S SYSTEM

In the proposed encryption scheme, a new hyper-chaotic system generated from Chen's chaotic system is used in key scheming, which is modeled.

$$- \begin{cases} \dot{x} = a(y-x)+w \\ \dot{y} = dx - xz + cw \\ \dot{z} = xy - bz \\ \dot{w} = yz + rw \end{cases} \qquad (1$$

Where x, y, z and w are state variables, a, b, c, d and r are parameters, when a =35,b =3,c =12,d =7 and 0.085G r G0.789,the system is hyper-chaotic. The hyper-chaos attractors are shown in Fig. 3.1.

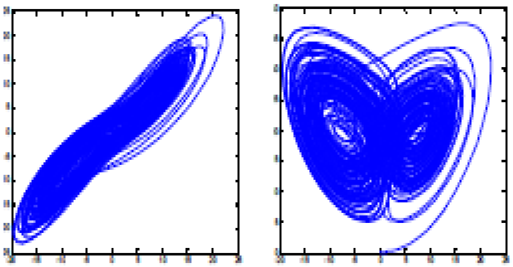

Fig. 3.1 Chen's hyper-chaotic attractor. (a) x-y plane, (b) x-z plane.

When a=35,b=3,c=12,d=7 and r=0.6. As the hyper-chaos has two positive lyapunov exponents, so the prediction time of a hyper-chaotic system is shorter than that of a chaotic system, as a result, it is safer than chaos in security algorithm

3.2 IMAGE ENCRYPTION BASED ON SHUFFLING MATRIX

Image value has strong correlations among adjacent pixels. In order to disturb the high correlation among pixels, an image shuffling matrix is used to shuffle the position of the plain-image. We assume that the dimension of the plain image is M × N ,the position matrix of pixel is ( ) , Pij(I) , i = 1,2...M, j = 1,2...M ,where Pij(I) stands for the grey value of the image. The procedure of shuffling image is described as follows:

Step1: For hyper-chaotic system, given initial value $x(i), y(i), z(i), w(i)$, after doing some iterations. For each iteration, we can get four values $x(i), y(i), z(i), w(i)$ where $i = 1,2...M$. represents the ith iteration of the hyper chaotic system. Then let

$$x'(i) = \mathrm{mod}((x(i)*10^{14}), M) \quad i = 1,2...M, x'(i) \in [0, M-1] \tag{2}$$

$$y'(i) = \mathrm{mod}((y(i)*10^{14}), M) \quad i = 1,2...M, y'(i) \in [0, M-1] \tag{3}$$

$$z'(i) = \mathrm{mod}((z(i)*10^{14}), M) \quad i = 1,2...M, z'(i) \in [0, M-1] \tag{4}$$

$$w'(i) = \mathrm{mod}((w(i)*10^{14}), M) \quad i = 1,2...M, w'(i) \in [0, M-1] \tag{5}$$

Step 2: Generate m by using the following formula:

$$m = \mathrm{mod}(x'(i), 6) \tag{6}$$

As $m \in [0,5]$,so from Table 1, we can select the corresponding group that are used to perform shuffling matrix. If m equals to the serial number of sequence of the group. For example, if $m = 2$, then choose $x'(i)$ and $w'(i)$.

Step 3: When we get $x'(i)$ and $w'(i)$, these data can be reordered in the form of{ $x'(i)$ },where $x'(i) \neq x(j)$, $j = 1,2...M$, if $i \neq j$. Then rearrange the row of matrix P_{ij} according to{ $x'(i)$ }, that is, move the $x'(1)$ to the first row, $x'(2)$ to the second row, thus a new image position matrix $P^{x}_{i,,j}$ is generated based on row transformation.

Step 4: For every row of the new matrix $P^{x}_{i,,j}$ we will shuffle the column position of the image. For $\{w'(i)\}$, these data can be reordered in the form of $\{w'(i)\}$, where $w'(i) \neq w'(j)$,if $i \neq j$. Then rearrange the data of every column for the first row of matrix $P^{x}_{i,,j}$ according to $\{w'(i)\}$,move the $w'(1)$ to the first column, $w'(2)$ to the second column, thus a new column transformation of the first row of matrix $P^{x,w}_{i}$ is generated.

Step 5: For the new $P^{x,w}_{i,j}$,go to Eq.(5),step 4 to do column transformation for the second row, till the last row transformation is finished, thus a new image total shuffling matrix $P^{x,w}_{i,j}$ is presented.

TABLE I. DIFFERENT COMBINATION OF STATES

Serial number	Combination of states
0	$\{x'(i), y'(i)\}$
1	$\{x'(i), z'(i)\}$
2	$\{x'(i), w'(i)\}$
3	$\{y'(i), z'(i)\}$
4	$\{y'(i), w'(i)\}$
5	$\{z'(i), w'(i)\}$

3.3 ENCRYPTION ALGORITHM DESIGN FOR THE HYPER-CHAOTIC CHEN'S SYSTEM

After we get the shuffling matrix Pij$^{x,\ w}$, the hyper-chaos is used to encrypt the shuffle image. The encryption scheme is based on the combination of state variables of the above hyper-chaotic system. One of the four variables are combined differently, which may produce four different combinations, which is given in table 2.

Step 1: The system on (1) is iterated for N_0 times.

Step 2: The hyper-chaotic system is iterated, as a result, four fraction will be generated. These decimal values are preprocessed firstly as follows:

$$\mathrm{x}(i) = \mathrm{mod}((abs(x(i)) - floor(abs(x(i)))) \times 10^{14}, 256) \tag{7}$$

$$\mathrm{y}(i) = \mathrm{mod}((abs(y(i)) - floor(abs(y(i)))) \times 10^{14}, 256) \tag{8}$$

$$\mathrm{z}(i) = \mathrm{mod}((abs(z(i)) - floor(abs(z(i)))) \times 10^{14}, 256) \tag{9}$$

$$\mathrm{w}(i) = \mathrm{mod}((abs(w(i)) - floor(abs(w(i)))) \times 10^{14}, 256) \tag{10}$$

Where $i = 1,2...M$ Represents the iteration of the hyper chaotic system. Where $abs(x)$ returns the absolute value of x . $Floor(x)$ returns the value of x to the nearest integers less than or equal to x , $\mathrm{mod}(x, y)$ returns the remainder after division.

Step 3: Generate $\overline{x}(i)$ using the following formula:

$$\overline{\mathrm{x}}(\mathrm{i}) = \mathrm{mod}(\mathrm{x}(\mathrm{i}), 4) \tag{11}$$

So $\overline{x}(i) \in [0,3]$ from Table 2, where $x(i)$ represents the ith iteration of the hyper chaotic system. We can select the corresponding group that is used to perform encryption operation. If $\overline{x}(i)$ equals to the serial number of sequence of the group. For example, if $\overline{x}(i) = 1$, then $y(i)$ is used to do encryption. According to the following formula:|

$$w(i) = p(i) \oplus y(i) \tag{12}$$

The process does not end until the set $P = \{p(1), p(2)...p(M \times N)\}$ is all encrypted. Then the encrypted pixel set $W = \{w(1), w(2)...w(M \times N)\}$ is written to the cipher-image.

The decryption algorithm is similar to the encryption algorithm. It is for the encrypted image, firstly, decrypt the image using hyper-chaotic system with the same parameters and initial values as that used in encryption, we will get the original image. These images will be show in the Fig.3.2.

TABLE II. DIFFERENT COMBINATION OF STATES OF HYPER-CHAOS

Serial number	Combination of states
0	$x(i)$
1	$y(i)$
2	$z(i)$
3	$w(i)$

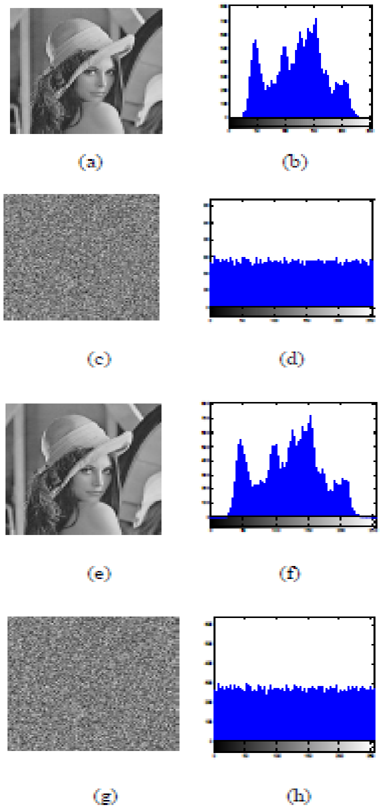
(a) (b)
(c) (d)
(e) (f)
(g) (h)

Fig.3.2 Image encryption and decryption experimental result. (a) Original image, (b) original image histogram, (c) ciphered image, (d) ciphered image histogram, (e) decrypted image, (f) decrypted image histogram, (g) decrypted image with different initial value, (h) error decrypted image histogram.

4. SECURITY ANALYSIS

Security is a major issue of a cryptosystem. A good encryption algorithm should be sensitive to the secret keys, and have large key space to resist all kinds of known attacks. Some security analysis has been performed on the proposed image encryption scheme.

A. Key space analysis
B. Key sensitivity test
C. Statistical Analysis
  ✓ Histogram
  ✓ Relativity Analysis
D. Anti-cutting test

Here we will discuss about Statistical Analysis
It is classified into two types. They are

- Histogram
- Relativity Analysis

1) Histogram

The figure is the Original image and the Encrypted image histogram, from Fig.3.2. We can see the encrypted image has uniform histogram, which means that it is a good encryption algorithm.

2) Relativity Analysis

To test the correction between two adjacent pixels in plain-image and ciphered-image, the following procedure was carried out. First, randomly select 512 pairs of two adjacent pixels from an image. Then, calculate the correlation coefficient of each pair by using the following formulas:

$$\mathrm{cov}(x,y) = E(x-E(x))(y-E(y)) \quad (13)$$

$$r_{xy} = \frac{|\mathrm{cov}(x,y)|}{\sqrt{D(x)}\sqrt{D(y)}} \quad (14)$$

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \quad (15)$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2 \quad (16)$$

$$\mathrm{cov}(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y)) \quad (17)$$

Where x and y are grey values of two adjacent pixels
in the image. These images were shown in figure 4.1:

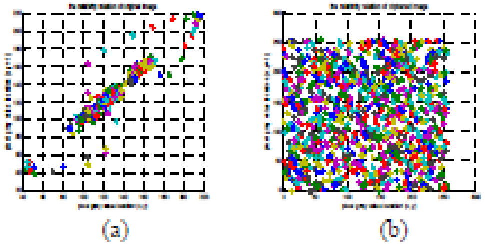
(a) (b)

Fig.4.1 Show the relativity relation of original and the encrypted image. (a) Relativity relation of original image, (b) relativity relation of encrypted image

5. CONCLUSION

In this paper, we proposed a new encryption algorithm called as hyper-chaos. It is used to change the location of the pixel from the image so we used image stegnography to hide the data/information then we apply this image to encryption by using hyper-chaos, the hackers cannot identify the data/information from the encrypted image. Because of image pixel was changed from the original image.

REFERENCES

[1] Moerland, T., “Steganography and Steganalysis”, *Leiden Institute of Advanced Computing Science*, www.liacs.nl/home/tmoerl/privtech.pdf

[2] Silman, J., “Steganography and Steganalysis: An Overview”, *SANS Institute*, 2001

[3] Jamil, T., “Steganography: The art of hiding information is plain sight”, *IEEE Potentials*, 18:01, 1999

[4] Wang, H & Wang, S, “Cyber warfare: Steganography vs. Steganalysis”, *Communications of the ACM*, 47:10, October 2004

[5] Anderson, R.J. & Petitcolas, F.A.P., “On the limits of steganography”, *IEEE Journal of selected Areas in Communications*, May 1998.

[6] Marvel, L.M., Boncelet Jr., C.G. & Retter, C., “Spread Spectrum Steganography”, *IEEE Transactions on image processing*, *8:08*, 1999

[7] Dunbar, B., “Steganographic techniques and their use in an Open-Systems environment”, *SANS Institute*, January 2002

[8] Artz, D., “Digital Steganography: Hiding Data within Data”, *IEEE Internet Computing Journal*, June 2001

[9] Johnson, N.F. & Jajodia, S., “Exploring Steganography: Seeing the Unseen”, *Computer Journal*, February 1998

[10] “Reference guide: Graphics Technical Options and Decisions”, http://www.devx.com/projectcool/Article/1997

[11] Owens, M., “A discussion of covert channels and steganography”, *SANS Institute*, 2002

[12] Johnson, N.F. & Jajodia, S., “Steganalysis of Images Created Using Current Steganography Software”, *Proceedings of the 2nd Information Hiding Workshop*, April 1998

[13] Venkatraman, S., Abraham, A. & Paprzycki, M., “Significance of Steganography on Data Security”, *Proceedings of the International Conference on Information Technology: Coding and Computing*, 2004