A New Steganography for Crypto Cover Files

Thanikaiselvan V.[1], Shubhanka Malpani[2], Megha Garg[2]
School of Electronics Engineering
VIT University, Vellore-632014, Tamilnadu, India.
thanikaiselvan@vit.ac.in[1]
shubhanka.malpani@gmail.com[2]
megha.garg812@gmail.com[2]

***Abstract*—Data security has become an indispensable part with rapid progress in digital technology. Secret information particularly needs to be handled with utmost care while transmitting through internet or any other media as well. Out of many techniques for securing data Cryptography and Steganography are the most efficient methods. The latter hides the data whereas the former encrypts the data. In this paper we propose a new methodology that incorporates both these techniques to facilitate robust data security. The cover image is first encrypted using Arnold algorithm, and then the secret data which could be a text file or binary data is hidden using LSB substitution and Pixel Value Differencing (PVD) to embed data adaptively in the encrypted cover image to give near full-proof security. In order to increase PSNR optimisation technique Optimum Pixel Adjustment Process (OPAP) has been used.**

Keyword- Steganography, PSNR, OPAP, LSB substitution, Pixel Value Differencing

I. INTRODUCTION

With the advent of internet and developments in information technology, issue of data security comes hand in hand. Secret information particularly sensitive information needs to handled with extreme care as its misuse can create havoc to the society. In order to ensure high level of security various Cryptography and Steganography techniques are used. While Cryptography makes the information unreadable to a hacker, Steganography aims at hiding secret data in an image, video or text file.

Image scrambling technology is easy to be realized but it is not in accordance with Kerchoffs's rules and its security is not very high. But when data is embedded in this scrambled image using Steganographic algorithms a high level of security is ensured ([1-10]). Steganography techniques involve hiding secret data in multimedia files like image, audio and video. It comes under the idea that if a certain feature is visible the point of attack becomes evident. Its main objective is to ensure robustness, high embedding capacity and undetectability [2]. The secret data is embedded in the Least Significant Bits (LSB) of the pixel values so as difference in the pixel value is not visible to the naked eye ([3]-[4]). Data embedding algorithms such as Pixel value differencing along with LSB substitution where difference in the pixel values is exploited to embed secret data makes the detection by hacker highly difficult. Data is embedded adaptively i.e. k bits are embedded based on the level in which the average difference falls ([5]-[9]).In order to further improve the image quality with low computational complexity optimization techniques such as Optimum Pixel Adjustment Process (OPAP) can be used. This method lowers the difference in the pixel values of original and stego.image thereby improving PSNR[3].

II. MATERIALS AND METHODS

In this paper we propose a new and secure data hiding technique by incorporating both cryptography and steganography techniques. The block diagram for our method is shown below in Figure 1. In this technique we have attempted to achieve high security by first encrypting the Cover file using a key and then embedding the secret data in the encrypted cover file using a steganography algorithm. For transmission the cover file is decrypted and then transmitted thereby hiding the very essence of any data hidden. As the steganography was performed on the encrypted cover file and any attempt to search for hidden data in the transmitted file results in garbage values. This ensures a highly secure way to transfer sensitive data as compared to using cryptography and steganography alone.

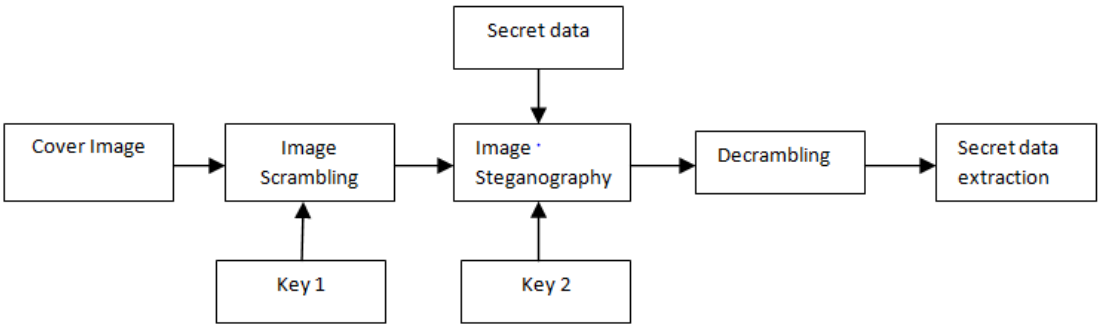


Fig 1. Block diagram of the process

At the receiver's end, the transmitted image undergoes the encrypting process again using the key. The secret data is then extracted from the image.

A. Image encryption algorithm using Arnold Transform Method

Arnold Transform is used as the encryption technique for the cover image. Arnold Transform is applied to the cover image in the spatial domain itself. The following matrix has to be used for encryption of the cover image.

$$\begin{bmatrix} p' \\ q' \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} p \\ q \end{bmatrix} \bmod N, \qquad p,q \in \{0,1,2\ldots N-1\} \tag{1}$$

Where $AT = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$ is the transformation matrix and N × N pixels is the size of the cover image. We have used the size of the cover image as 512 × 512 pixels. Also, (p, q) are the pixel location of the original image, (p ', q') as a pixel location of the scrambled image. The multiplication i.e. the location of one pixel is changed multiple times. The number of times it is changed is the key for the encryption process and is stored as the number of iterations.

The Arnold Transform method of encryption is periodic in nature. It is robust, quick and has high confidentiality. After Arnold transformation of the image, pixel's location of the image in space is completely changed, but it does not change the value of image pixel, so the image histogram is the same. In order to change the pixel values chaotic sequence can be used. [1]

B. Steganography

Now that we have encrypted the cover image, we embed the secret data inside it. The secret data can be a text file, which has been converted completely into binary. In this paper we have only embedded binary data.

1) *LSB Substitution Method:* The technique followed in this paper is of LSB substitution wherin the secret data is embedded into encrypted pixel by changing the LSB of the pixel so that it holds the bits of the secret data. Let C be the original 8-bit grayscale cover-image of 512 × 512 pixels represented as

$$CI = \{p_{ij} \mid 0 \le i,j \le N, p_{ij} \in \{0,1,\ldots.,255\}\} \tag{2}$$

Let M be the n-bit secret message represented as :

$$Msg = \{m_i \mid 0 \le i < n, m_i \in \{0,1\}\} \tag{3}$$

Suppose that the n-bit secret message M is to be embedded into the k (where k can be 1,2,3,4) rightmost LSBs of the cover-image C. This n-bit binary message is converted to its decimal value by combining k bits together (z_i). The substitution is done by using the following equation:

$$p_{ij}' = p_{ij} - \bmod(x_{ij}, 2^k) + z_i \tag{4}$$

The new pixel value has the secret data embedded in the last k bits of its pixel value. As the data is hidden only in the LSB, the pixel value does not change significantly and is not visible to the human eye. ([3]-[4])

2) *Pixel Value Differencing:* The encrypted cover file is taken and is divided into sub-blocks of 2 × 2 pixels. We use an adaptive method to embed the data, i.e. the number of bits embedded in each pixel is not the same. It is in accordance with a condition. This makes the data embedding procedure even more robust.

A threshold value Th is taken for the embedding process. The range of Th is $2^{kl} \le Th \le 2^{kh}$, where kl and kh are the number of bits embedded in the lower threshold and upper threshold respectively.

A sub-block of 2 × 2 pixels is taken from the image. Out of the 4 pixels, we find the minimum (P_{min}). Then using the formula in (3) we find the value of delta. According to the relation of Delta with Th, we either go for higher level embedding (if Delta>Th) or lower level embedding (if Delta<Th).

$$Delta = \frac{\sum_{i=1}^{4}(P_i - P_{min})}{3} \quad (5)$$

where, P_1, P_2, P_3, P_4 are the four pixel values of the 2 × 2 sub-block.

Using this method for Steganography gives higher embedding capacity and better image quality. The secret data is secure as the key (i.e. the threshold value Th, kl and kh) needs to be known for the extraction of information. Without the key there is no way of extracting the data. ([4]-[5])

C. Optimum Pixel Adjustment Process (OPAP)

This method is used to enhance the quality of the stego image obtained by LSB substitution. This method is used to reduce the visible difference between the stego image and the encrypted cover image. Let a, a', a" be the corresponding pixel values of the i^{th} pixel in the encrypted cover-image.

Let $\partial = a - a'$ (6)

be the embedding error between stego image and the cover image. The pixel value a' is obtained by replacing the LSB of the cover image with the secret data. So the range of ∂ is $-2^k < \partial < 2^k$.

If $\partial > 2^{(k-1)}$ and $2^k < a' < (256-2^k)$ then we change the value of the pixel as $a'' = a' + 2^k$

If $\partial < -2^{(k-1)}$ and $2^k < a' < (256-2^k)$ then we change the value of the pixel as $a'' = a' - 2^k$.

This technique is used to improve the Peak Signal to Noise Ratio (PSNR) so as to get an image of better quality. [3]

D. Decryption

Before the transmission of the stego image, the image is decrypted so as to get the original cover image. This is done so that the hackers do not suspect the presence of any information in the image. If the hackers try to get information then it will result in garbage values as the decryption process has changed the sequence in which data was stored. The following matrix is used for the decryption process; it is the inverse of the Arnold Transform Matrix:

$$\begin{bmatrix} p' \\ q' \end{bmatrix} = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} p \\ q \end{bmatrix} \bmod N, \qquad p,q \in \{0,1,2 \ldots N-1\} \quad (7)$$

where $AT = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}$ is the inverse transformation matrix and N×N is the size of the cover image. We have used the size of the cover image as 512×512.Also, (p, q) are the pixel location of the scrambled image, (p', q') as a pixel location of the descrambled image. The multiplication i.e. pixel location change is performed as many times as when it was scrambled. The decrypted image is now ready for transmission. [1]

E. Stegextraction

The first step of the stegextraction process is to scramble the received image again using Arnold Transform and the key as described above. Now the image is of the form in which the data was embedded. The key contains the information about the Threshold Th, and the kl and kh, ie the number of bits that were embedded in the different levels. A sub-block of 2 × 2 pixels is taken from the image. Out of the 4 pixels, we find the minimum and maximum. Then using the formula in (8) we find the value of delta. According to the relation of Delta with Th, we either extract kh bits of information (if Delta>Th) or kl bits (if Delta<Th).

$$Delta = \frac{\sum_{i=1}^{4}(P_i - P_{min})}{3} \quad (8)$$

where, P_1, P_2, P_3, P_4 are the four pixel values of the 2 × 2 sub-block.

Thus, the information that was embedded has been successfully extracted.

III. PROPOSED METHODOLOGY

A. Embedding Algorithm

Step 1: Read a 512×512 image as Cover Image

Step 2: Apply Image scrambling Algorithm using Arnold Transform Method

Step 2.1: Calculate the size of the image and store it in N, ie. N=512

Step 2.2: Run two loops for the rows and columns. Find the new location of the pixel using the equation (1).

Step 2.3: The pixel value of the original pixel location is assigned to the new pixel location.

Step 2.4: Steps 2 and 3 are repeated for number of times we want to change the location of each pixel. This number of iterations serves as the key for the encryption process. [1]

Step 2.5: Supply the scramble image for the next step for Steganography

Step 3: Apply Pixel value Differencing based Steganography to the Image

Step3.1: A threshold value Th is taken for the embedding process. The range of Th is $2^{kl} \leq Th \leq 2^{kh}$, where kl and kh are the number of bits embedded in the lower threshold and upper threshold respectively.

Step 3.2: The entire cover image is divided into sub-blocks of 2× 2 pixels. The sub-block is extended from the top left corner rightwards.

Step 3.3: Let the 4 pixel values in the sub block beP_1, P_2, P_3 and P_4. The minimum of these 4 pixels is found and stored in P_{min}.

Step 3.4: Now we find Delta for the 4 pixels using the equation (5).

Step 3.5: If Delta>Th then kh number of bits of information are embedded in each pixel of sub-block using the LSB substitution method.

Step 3.6: If Delta<=Th, then we check for error block condition:

$P_{max} - P_{min} > 2*Th + 2$ (9)

If the sub-block belongs to the error block then we do not embed secret data in this block. The reason being that any change in the pixel value of such a block causes it to be easily noticeable.

If the sub-block does not belong to the error block then we embed kl bits of information in each pixel the sub-block using LSB substitution

Step 3.7: Execute OPAP for improving the PSNR.

Step 3.8: Steps 3.3 to 3.7 are repeated till all the pixels have undergone this process. ([4]-[5])

Step 3.9: Generate a Steganography image.

Step 4. Apply Descrambling algorithm with Arnold Transform, The decrypted image is now ready for transmission.

B. Extraction Algorithm

The first step of the stegextraction process is to scramble the received image again using Arnold Transform and the key as described above. Now the image is of the form in which the data was embedded.

Step 1: The entire image is divided into sub-blocks of 2× 2 pixels. The sub-block is extended from the top left corner rightwards.

Step 2: Let the 4 pixel values in the sub block beP_1, P_2, P_3 and P_4. The minimum of these 4 pixels is found and stored in P_{min}.

Step 3: Now we find Delta for the 4 pixels using the equation (5)

Step 4: If Delta>Th then kh bits of information are extracted from each pixel using mod(P_i,2^{kh})

Step 5: If Delta<=Th, then we check for error block condition using equation(8)

If the sub-block belongs to the error block then we do not extract information from this block. If the sub-block does not belong to the error block then we extract kl bits of information from each pixel using mod(P_i,2^{kl}).

Step 6: The extracted information is concatenated together to form one string of binary data.

Step 7: Steps 2 to 6 are repeated till all the pixels have undergone this process.

Thus, the information that was embedded has been successfully extracted.

IV. Results and Discussion

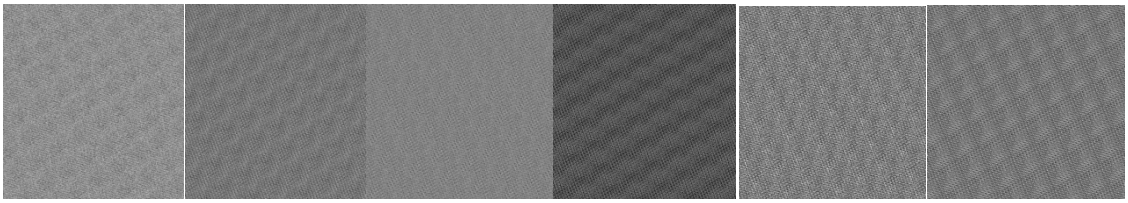
Experiments using Matlab software have been conducted on 6 grayscale images with size 512 × 512. PSNR values along with number of bits embedded in each scrambled image has been calculated. A text file has been used as secret data which is converted from its ASCII value to binary to make it suitable for embedding in the scrambled cover image. The peak signal to noise ratio (PSNR) is used to evaluate the quality of the scrambled stego image. For an A × B grayscale image PSNR is calculated as:

$$PSNR = 10 * log_{10} \frac{255*255*A*B}{\sum_{i=1}^{A}\sum_{j=1}^{B}(p_{i,j}-q_{i,j})^2} \text{(dB)} \quad (10)$$

where p_{ij} and q_{ij} denote the pixel values in row *i* and column *j* of the cover image and the stego.image. [6]

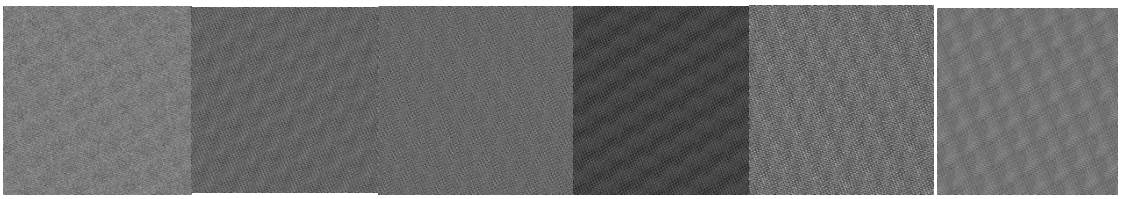
(a) (b) (c) (d) (e) (f)

Fig 2. 6 cover images with size 512*512: (a) Baboon (b)Peppers (c) Lena (d) Girl (e) Barbara (f) Elaine



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

Fig 3. 6 Scrambled images with size 512*512: (a) Baboon (b)Peppers (c) Lena (d) Girl (e) Barbara (f) Elaine



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

Fig 4. 6 Stego images with size 512*512: (a) Baboon (b)Peppers (c) Lena (d) Girl (e) Barbara (f) Elaine

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

Fig 5. 6 descrambled images with size 512*512: (a) Baboon (b)Peppers (c) Lena (d) Girl (e) Barbara (f) Elaine

TABLE I

Image	**Th=5, 2-3(kl-kh)**		**Th=6 2-3(kl-kh)**		**Th=12 2-4(kl-kh)**		**Th=18 3-5(kl-kh)**	
	Capacity	**PSNR**	**Capacity**	**PSNR**	**Capacity**	**PSNR**	**Capacity**	**PSNR**
Baboon	786200	46.7283	786032	46.7421	1044672	40.8090	1297828	34.5417
Peppers	786244	46.5332	786164	46.5262	1044872	40.1590	1297980	33.9131
Lena	785600	46.7542	785472	46.7503	1040920	40.9344	1291552	34.2050
Girl	785796	46.1458	785420	46.1471	1038256	40.3377	1282908	34.1271
Barbara	786336	46.5573	786304	46.5543	1046840	39.8757	1305536	32.6936
Elaine	786244	46.7706	786128	46.7657	1044448	40.8266	1296608	34.5345

TABLE III

At threshold Th=5, 2-3 (kl-kh)			
Image	**No of bits**	**PSNR**	**PSNR with OPAP**
Lena	785600	42.7434	46.7542
Peppers	786244	42.7232	46.5332
Baboon	786200	42.7095	46.7283
Girl	785796	43.4148	46.1458
Barbara	786336	42.7899	46.5573
boat	786352	49.7436	49.7944

Stego images created by the proposed method from fig (2-5).Fig 2. Shows 6 grayscale images of size 512×512 which are the original images .Fig.3 shows result after scrambling the original image using Arnold algorithm. Here we have run the algorithm 10 times however it can be used n number of times. Fig.4 shows scrambled images with data embedded inside using PVD and LSB substitution .Fig.5shows descrambled stego images transmitted through a suitable channel. As can be seen no significant difference can be noticed in original images and descrambled stego images.

In Table III various threshold values as Th=5, 6, 12, 18 with various kl-kh values have been taken and PSNR value and bit embedding capacity has been calculated. For example at Th=5 ,if Delta<=Th 2bits are embedded and at Delta>Th 3 bits are embedded in a 4 pixel block. It can be observed that with the increase in threshold value PSNR reduces but the embedding capacity (in bits) increases at the same time. In Table II after applying OPAP to 6 grayscale images of size 512×512 at threshold value Th=5 the increase in PSNR value is quite significant with capacity (in bits) remaining the same. Various threshold values can be taken and tested for the same.

V. CONCLUSION

In this paper we have proposed a steganographic data security algorithm for ensuring high level of data security. First the cover image has been scrambled using Arnold algorithm and data has been embedded using PVD and k-bit LSB substitution in the scrambled image to incorporate high randomness. It has been observed that as as the threshold value goes up the embedding capacity increases significantly but there is a decrease in PSNR value that is the picture quality degrades. After using optimization technique OPAP the increase in PSNR values is significant. This proposed method majors in terms of embedding capacity and making the secret data fool proof. However there is a tradeoff between PSNR and embedding capacity at high threshold values. Improving the same will be our future endeavor.

REFERENCES

[1] Yuanmei Wang Tao Li".Study on Image Encryption Algorithm Based on Arnold Transformation and Chaotic System"2010 International Conference on Intelligent System Design and Engineering Application

[2] Abbas Cheddad, Joan Condell, Kevin Curran, Paul Mc Kevitt, Digital image steganography: Survey and analysis of current methods ,*Information Sciences, 2011*

[3] Chan CK, Chen LM. Hiding data in images by simple LSB substitution, *Pattern Recognition*, 2004;**37** : 469–474.

[4] Wu HC, Wu NI, Tsai CS, Hwang MS. "Image steganographic scheme based on pixel-value differencing and LSB replacement methods", *Proc. Inst. Elect.Eng., Vis. Images Signal Process* 2005

[5] Xin Liao a, Qiao-yan Wena, Jie Zhang b "A steganographic method for digital images with four-pixel differencing and modified LSB substitution" J. Vis. Commun. Image R. 22 (2011) 1–8

[6] Thanikaiselvan V, Arulmozhivarman P, Rengarajan Amirtharajan, John Bosco Balaguru Rayappan "Horse Riding & Hiding in Image for Data Guarding" Original Research Article *Procedia Engineering, Volume 30, 2012, Pages 36-44*

[7] Katzenbeisser S, Petitcolas FAP. "Information Hiding Techniques for Steganography and Digital Watermarking, Artech House,Norwood", MA, 2000

[8] Thien CC, Lin JC. A simple and high-hiding capacity method for hiding digit-by-digit data in images based on modulus function, *Pattern Recognition, 2003;* ***36: 2875–2881***

[9] Yang CH, Weng CY. A steganographic method for digital images by multipixel differencing, *in: Proceedings of International Computer Symposium,Taipei, Taiwan, R.O.C., 2006; p. 831–836.*

[10] H.J. Highland, Data encryption: a non-mathematical approach, Comput Secur. 16 (1997) 369-386.