LSB Algorithm based on Support Vector Machine in Digital Image Steganography

Hanizan Shaker Hussain[1], Roshidi Din[2], Aida Musthapa[3], Fawwaz Zamir Mansor[3]
[1]*Kuliyah Teknologi dan Multimedia Kreatif, Kolej Univerisiti Insaniah, 09300, Kuala Ketil, Kedah, Malaysia*
[2]*School of Computing, UUM College of Arts and Sciences, Universiti Utara Malaysia, 06010 Sintok, Kedah, Malaysia*
[3]*Faculty of Computer Science and Information Technology, Universisti Tun Hussein Onn Malaysia, 86400 Parit Raja, Batu Pahat, Johor Malaysia*
drhanizan@insaniah.edu.my

***Abstract*— The importance of information security in protecting data and information has increased due to the increased use of computers and the Internet. It similar with the one of exciting subfields of information security called information hiding. Information hiding is a technology where the secret-messages are hidden inside other files (e.g image files). One of the areas that are popular now applying this technology is digital image steganography (image steganography). In this paper proposed StegaSVM-Shifted LSB model that has been proposed that utilize HVS and embedding technique through Shifted LSB showed a good performance. This can be seen when PSNR record high value, where it displays a good quality cover-image.**

***Index Terms*— Discrete Consine Transformation; Image Steganography; Support Vector Machine.**

I. INTRODUCTION

Digital image steganography (image steganography) is a field in information hiding, besides watermarking, that hides and secures secret-messages written inside an innocuous-looking cover-image file. As a sub-field in information security, it is currently used to secure transmission and information storage in many fields like in the military and medical fields.

The word steganography is derived from the Greek words "stegos" meaning "cover" and "grafia" meaning "writing" defining it as "covered writing" [1].The first steganographic algorithm was developed in ancient Greece around 440 B.C. involved the shaving of a slave's head, then a tattoo was inscribed on the scalp. When the slave's hair had grown back and hidden the message, the slave was sent to warn of the Persians' impending invasion. The recipient once again shaved the slave's head and retrieved the important warning. Modern steganography is being used all this while especially in military referred as transmissions security [2], [3], [4] and in medical field to secure the confidential patient's information [4], [5].

In designing steganography algorithms, three requirements are generally considered namely imperceptibility, robustness and payload capacity [5], [6]. Nevertheless, in image steganography, the most important requirements are imperceptibility and robustness [7], [8]. These two requirements are important to resist especially two kinds of statistical attacks that is passive and active steganalysis. Lately many researchers concentrates on passive steganalysis,the attacks only takes actions when the stego-image is found suspicious [9], [10]. Meanwhile, active steganalysis is the attacks that attempt to foil all possible covert communications [11], [12].Therefore, this study used StegaSVM-Shifted LSB model is introduced which takes into account. Then, the important factor is taken into consideration in image steganography which is randomizing embedding, HVS and also the ability to recover the right secret-message in the extracting process. It design consist two main models, StegaSVM classification and StegaSVM-Shifted LSB. In StegaSVM classification, the classified cover-image will be utilized and the non-smooth area, the most appropriate and imperceptible is preferred to be used during the embedding process. StegaSVM classification also will be applied in the extracting process in order to extract the right secret message.

There are a lot of algorithms have been proposed and developed till this day in image steganography. But many of them have failed to attain at least the main purposes of steganographic algorithm which is to keep the existence of the secret-message secret [9], [10] and secret-message is robust enough to resist the passive steganalysis (e.g. image processing operation) [13], [14].

In fact, many image steganographic algorithms have faced problems where the secret-message is not only easy to be detected but also distorted at the end of the steganographic process.The main concernthat contribute to this problem is the non-random changes on a cover-image that constantly occurred after the embedding process [15]. Among the causes that lead to the non-random changes is the exploitation of the non-random embedding algorithm [16]. The non-random changes also distinctively occured if the secret-message is embedded in the inappropriate area like smooth areas (e.g. plain or bright areas)on the cover-image [17], [18]. Another cause for the non-random changes is the payload capacity that exceed the number of allowed bits in the cover-mage [18], [20]. All of these factors have created the cover-image easily to be detected and manipulated by passive steganalysis. Therefore, this paper focus on development of algorithm in image steganography specifically in LSB algorithm.

II. RELATED WORK

Nowadays, steganography has grown to be as an emergent topic. There are four types of steganography, being image steganography, text steganography, video steganography and audio steganography and the type of steganography actually depends on what kind of carrier file that is used [5], [15]. Figure 1 below shows the types of steganography. A

good indicator of the growing interest in this subject is the number of special sessions that have been held in recent conferences and highlighted in several scholar researches [21], [22].

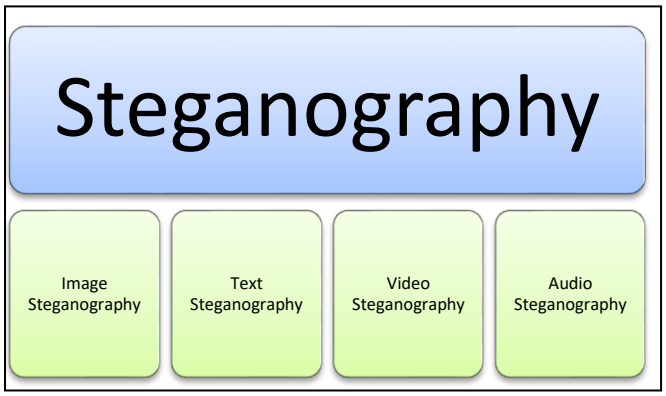

Figure 1: Types of major steganography categories

Image steganography is a technique to hide secret-messages into a digital image. This steganography technique exploits the weakness of the human visual system (HVS). HVS cannot detect the variation in luminance of color vectors at higher frequency side of the visual spectrum. An image can be represented by a collection of color pixels. The individual pixels can be represented by their optical characteristics like 'brightness', 'chroma' etc. Each of these characteristics can be digitally expressed in terms of 1s and 0s. This technique can be directly applied on digital image in bitmap format as well as for the compressed image format like JPEG. In JPEG format, each pixel of the image is digitally coded using discrete cosine transformation (DCT). The LSB of encoded DCT components can be used as the carriers of the secret-message.

A. Image Steganography

There are many classifications of image steganographic techniques. But most are classified according to the image domain i.e. spatial and transform techniques [5,15]. Spatial technique embeds the secret-message into the pixels of the cover-image directly, while transform technique embeds the secret-message into the cover-image by modifying the coefficients in a transform domain, such as the DCT.

The LSB based technique as described earlier is based on the bit-replacing while BPCS technique hides secret-message by the way of block-replacing. There are several reasons why this technique is still applied by steganographers among others as great capability of secret-message, fine concealment and easy realization. Among the several tools of this kind of technique are S-Tools, Hide and Seek, Hide4PGP and Secure Engine Professional. While files such as BMP, GIF, PNG images and WAV audio area lossless format used as a cover-image by these tools.

Another technique, transform technique embeds secret message in the transform coefficients typically to achieve imperceptibility and robustness. The technique mainly includes JSteg, F3, F4, F5 [1], [16] and many more of this kind of techniques can be found later in this section. The representative tools include JSteg shella 2.0, JPHS, F5, outguess, Steganos Security Suite 6.0 [17].

Although there are many ways for embedding as discussed above, basically, most image steganographic techniques for embedding consist of these two steps:

i. Firstly, the cover-image is analyzed and the imperceptible and insignificant bits are identified. It is assumed that changing these bits will not make observable changes to the cover-image.
ii. Secondly, identified bits are then substituted by the secret-bits to create an altered cover-image called stego-image.

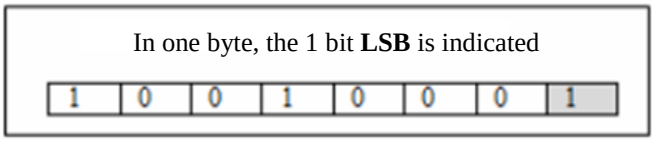

Figure 2: Least significant bit is indicated on the far right

In fact, in image steganography, the most popular and widely used technique to hide data is the usage of least significant bit (LSB) [1], [22]. LSB has been extensively utilized to embed secret-messages into a cover-image in a spatial and DCT domain as well [23]. In the literature of image steganography, the LSB algorithm has shown many improvements over time, making it more robust and preferred algorithm by researchers of steganography.

The way in which this technique to hide secret-message is achieved by modifying the least significant bit layer of a cover-image. The last bit of the byte is frequently selected as the least significant bit because of the impact of the bit to the minimum distortion of stego-images [24].

In this study the concentration is more on transform domain and in DCT domain specifically. DCT domain techniques hide secret-messages in LSB of the DCT coefficients of the cover-image which makes them more robust to some image manipulations, such as compression, cropping, and rotating. While they are more robust to various kinds of signal processing, they remain imperceptible to the human visual system (HVS). Meanwhile, the discovery of the LSB embedding mechanism is actually a big achievement to steganography even though it is perfect in not deceiving the HVS, its weak resistance to attacks has left researchers wondering where to apply it next it until they have successfully applied it within the DCT domain [5].

B. Support Vector Machine

Support vector machine (SVM) is a probabilistic learning theory that given a set *S* of labeled training points:

$$S = \{(x_i, y_i) \mid x_i \in R^n, y_i \in \{-1,1\}, i = 1...m\} \quad (1)$$

m = total number of attributes

where x_i stands for input vector *i* and y_i is the class label, positive and negative. SVM can generate a separation hyperplane H that separates the positive and negative examples. Since SVM has the high generalization ability to separate data into two classes, thus it is naturally suitable to classify the cover-image. If any point *x* which lies on the hyperplane must satisfy $w \cdot x + b = 0$, where *w* is normal to the hyperplane and *b* is the bias. Lastly, the optimal hyperplane H: $w_0 .x + b_0 = 0$ can be determined by

$$w_0 = \sum_{i=1}^{m} \alpha_i y_i x_i \quad (2)$$

where α_i and b_0 are lagrange multipliers and bias that determined by SVM's training algorithm.

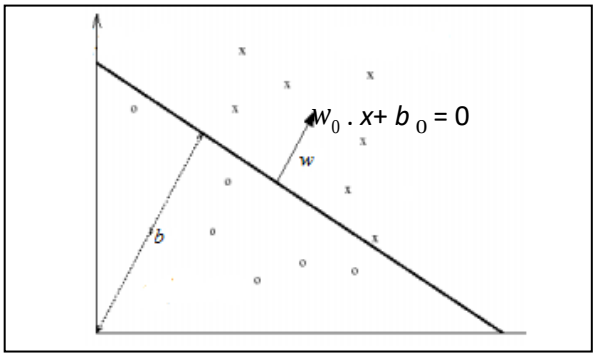

Figure3: Optimal hyperplane H

Figure 3 showed the hyperplane H which stretches from the x axis tothe y axis that separates 'x' and 'o' data points. The same figure also shows the direction of hyperplane H represented by *w* and *b* shows its position in space. After the training of SVM is completed, H is thus determined, then any data *x* will be classified according the sign of the decision function. The decision function is defined as.

$$d(x) = \mathrm{sgn}(\sum_{i=1}^{m} \alpha_i y_i K(x_i x) + b_0 \quad (3)$$

where $K(x_i x)$ is the kernel function which maps the training samples to a higher dimensional feature space. Two kinds of kernel functions are commonly used in SVM, the radial bias function (RBF) and polynomial to classify the cover-image in the digital image steganography (Hamel, 2009; Meng *et al.*, 2008; Kecman, 2001).Generally, the original input space can always be mapped to some higher dimensional space where the training set is separable.

C. SVM Work in image Information Hiding

In statistical learning theory, the SVM has been developed for data classification and prediction and it is used in wide range applications such as character recognition [25], text categorization, face detection in images and information hiding in images [26].

The latest development in SVM such as [27], [26] and [17] moves toward utilizing SVM in DCT domain. The DCT domain is recommended in hiding messages because of the minimum degradation of stego-image [23]. They not only exploit the DCT domain but also use the grayscale images since it is more robust and secured to be a cover-image [11], [23]. Even if the human visual system (HVS) can be exploited, it would preserve the stego-image's imperceptibility. It is because the imperceptibility is closely related to the HVS characteristics [26]. In [27] has exploited HVS when classifying the cover-image based on the luminance and texture values. He also utilizes the DCT coefficients of middle frequency for secret-message embedding as the middle frequency have similar magnitudes and is relatively more secured whereas [26] utilizes the types of textures in exploiting the characteristic of HVS. SVM together with associated fuzzy membership value of each pixel are then used to classify the cover-image into smooth and non-smooth area for embedding. Meanwhile, in [17] uses brightness and edge values to classify the cover-image to fit the HVS. She then exploits the non-smooth area rather than the smooth area for embedding to preserve the imperceptibility of stego-image. In [28] and [29] are still attempting to use pixel intensity to classify the color cover-image but this time they are exploiting the transform domain for the embedding.

III. STEGASVM-SHIFTED LSB DEVELOPMENT

In the StegaSVM-Shifted LSB Development phase, three developmental activities will be carried out, that are, stage development for the StegaSVM classification, embedding and also extracting. Activities are based on three key developments that at the end of this phase, three algorithms are developed which StegaSVM classification, Shifted LSB embedding and Shifted LSB extracting algorithms in Figure 4 as follows.

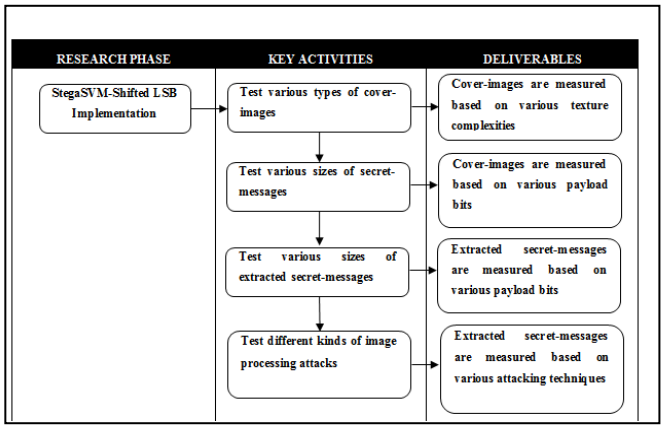

Figure 4: StegaSVM-Shifted LSB implementation phase

In this phase, the StegaSVM-Shifted LSB implementation, there are four activities involved that are. It is to test various types of cover-images, various sizes of secret-messages, various sizes of extracted secret-messages and different kinds of image processing attacks. But to ease the implementation and also the understanding of this experiment, those tests were divided to imperceptibility and robustness requirements. For both activities; to test various types of cover-images and various sizes of secret-messages put under imperceptibility requirement. While the remaining two, various sizes of extracted secret-messages and various kinds of image processing attacks is an experiment that is used to measure the rate of robustness. The final result is to test various types of cover-images, various sizes of secret-messages and quality of the cover-images is measured based on various texture complexities. Then, the quality of the cover-images is also measured based on various payload bits for the second test. Thirdly, from the third test the extracted secret-messages is measured based on various payload bits and finally the extracted secret-messages is measured based on various attacking techniques.

A. StegaSVM-Classification

StegaSVM classification stage is the first stage in the development phase of the StegaSVM-Shifted LSB tool. There are several important activities in this development

phase that have been highlighted here which are image features extraction.

1) SVM Training Dataset Construction

For the purpose of preparing a data in SVM format besides ImageJ application, one JAVA program, *SVMLabelDataset* has been developed to put in order the data (refer Appendix A). Figure 5 below shows several coding excerpts that are used to extract image features from cover-image i.e. *Lena.bmp*.

```
public class SVMLabelDataset
{
    public SVMLabelDataset()throws Exception
    {
        try
        {
            FileReader reader = new FileReader("lena_luma.txt");
            BufferedReader in = new BufferedReader(reader);
            String inData=null;

            FileReader reader2 = new FileReader("lena_edge.txt");
            BufferedReader in2 = new BufferedReader(reader2);
            String inData2=null;

            FileReader reader3 = new FileReader("lena_entropy.txt");
            BufferedReader in3 = new BufferedReader(reader3);
            String inData3=null;


            FileWriter writer1 = new FileWriter("lenatest_label.txt");
            PrintWriter out1 = new PrintWriter(writer1,true);

            FileWriter writer2 = new FileWriter("lenatest_data.txt");
            PrintWriter out2 = new PrintWriter(writer2,true);

            FileWriter writer3 = new FileWriter("lenatrain_label.txt");
            PrintWriter out3 = new PrintWriter(writer3,true);

            FileWriter writer4 = new FileWriter("lenatrain_data.txt");
            PrintWriter out4 = new PrintWriter(writer4,true);
```

Figure 5: Excerpt coding to extract image features

At the same time, Figure 6 shows several coding excerpts to construct dataset in SVM format from *Lena.bmp* image.

```
for(int i=0;i<256*256;i++)
{
    if(((inData=in.readLine())!=null) && ((inData2=in2.readLine())!=null)&& ((inData3=in3.readLine())!=null))
    {
        StringTokenizer str = new StringTokenizer(inData," ");
        StringTokenizer str2 = new StringTokenizer(inData2," ");
        StringTokenizer str3 = new StringTokenizer(inData3," ");

        int value = Integer.parseInt(str.nextToken());
        int value2 = Integer.parseInt(str2.nextToken());
        int value3 = Integer.parseInt(str3.nextToken());

            if((value < 114)&& (value2 < 84)&& (value3 < 128))
            {
                out1.println("-1");//smooth area
                out2.println(value+"    "+value2+"    "+value3);
            }
            else
            {
                out1.println("1");//non-smooth area
                out2.println(value+"    "+value2+"    "+value3);
            }
```

Figure.6. Excerpt coding to construct dataset in SVM format

In addition, the selection of the optimal threshold values for each of the cover-image that will be used in every experiment is equally important in this development stage. Threshold values for each of the cover-image are used to determine the class label either smooth (-1) or non-smooth (1) later in stegaSVM classification. To achieve this purpose, ImageJ application (ImageJ) again is used. ImageJ will apply its own thresholding method that automatically set the threshold levels based on an analysis of the histogram of the current cover-image (Schneider *et al.*, 2012).

2) SVM Classification Algorithm

The first step in the StegaSVM classification algorithm is to read a grayscale cover-image. Then, from the input cover-image, image features extracted consisting of luminance, edge and entropy. Next step, from those values, SVM training dataset will be constructed.

The next activity is the consideration on types of kernel that wants to be used. In this research, the RBF kernel has been chosen to be applied based on the following formula $K(x,y)= e^{-\gamma\|x-y\|^2}$. After that, the cross-validation technique will be used to determine the parameters (i.e. penalty parameter (*C*) and Gamma (γ)) that are suitable for this research. After the cross-validation technique has been done, the SVM training set that has applied the best best parameter choosen from the previous step can be executed. The final result is the trained stegaSVM function *f*(x). *f*(x) function is the function that is going to be used in Shifted LSB Embedding process. The step is for stegaSVM classification algorithm is summarized as follows:

1. Read cover_image
2. Extract image-features from cover-image
3. Construct the SVM training dataset
4. Select the SVM kernel i.e. RBF, $K(x,y)= e^{-\gamma\|x-y\|^2}$
5. Use cross-validation technique to find the best parameter, penalty parameter C and gamma value γ.
6. Use the best parameter C and γ to train the whole training set to generate the trained SVM function f(x)

By having the StegaSVM classification algorithm, then the suitable SVM classification model with image steganography is determined. The model is essential to be determined because it can affect the embedding and extracting process later on especially in determining a few important SVM parameters for instant *C* and γ values. From the experiments conducted, we can conclude that the best value of the *C* is 30 while the best γ value is 0.5 for this suggested image steganographic model case.

3) Shifted LSB Embedding Algorithm

After going through the process of DCT transform, quantization and also zig-zag scan, a cover-image goes through about the next stage SVM classification and embedding process called Shifted LSB embedding. The algorithm is called StegaSVM-Shifted LSB embedding algorithm that is occurred according to the following steps.

```
1.  Read secret-message
2.  Convert secret-message into a sequence of secret-bits
3.  Encrypt and compress the secret-bits using a secret key
4.  Divide the cover-image to obtain non-overlapping 8x8 blocks
5.  Transform the cover-image into DCT domain and then quantize
6.  Get a random location of DCT coefficients on cover-image using PRNG using
    a secret key
7.  Get DCT coefficient of cover-image
8.  Determine smooth and non-smooth locations from the trained StegaSVM
    function, f(x)
    8.1   IF DCT coefficient is non-smooth THEN
          8.1.1 Embed
                Go to Step 9
    8.2   ELSE
          8.2.1  Get embedding strength
          8.2.2  Get DCT coefficent of 8-neighbors' LSB & Counter=+1
          8.2.3  IF DCT coefficient is non-smooth THEN
                 Go to Step 8.1.1
                 ELSE IF Counter<=8 THEN
                        Get Neighbors embedding strength
                        Go to Step 8.2.3
                      ELSE
                        Find DCT coefficient with highest embedding
                        strength
                        Go to Step 8.1.1
9.  IF secret-bit ≠ EOF
        Go to Step 7
    ELSE
        Go to Step 10
10. Store the resulting cover-image as stego-image
```

From the steps of StegaSVM-Shifted LSB embedding algorithm mentioned above, it can be summarized as follows; first the secret-message that has been input will be converted to a sequence of secret-bits. Then, using PRNG with secret key *K*, secret-bits will be encrypted and compressed. After that, there will be an input of cover-image and divided to non-overlapping 8x8 blocks. Next, this cover-image will be transformed to discrete cosain transform (DCT) domain and also quantized. The next step is to get the embedding location randomly from the DCT coefficeints using the (PRNG). PRNG method is commonly used and has a good potential in research that involves SVM technique and data embedding (e.g. Jain and Tiwari, 2011; Tsai *et al.*, 2010; Wu *et al.*, 2008; Fu, 2005).

4) Shifted LSB Extracting Algorithm

The extraction process occurs starting with SVM classification to classify cover-image for the determination of SVM label to the smooth and non-smooth areas. Then the process of extraction of secret-bits will take place where the steps are as follows:

```
1.  Read stego-image
2.  Divide the stego-image to obtain non-overlapping 8X8 blocks
3.  Transform the cover-image into DCT domain and then quantize
4.  Get a random location of DCT coefficients on stego-image using PRNG with a
    secret key
5.  Get DCT coefficient of stego-image
6.  Determine smooth and non-smooth locations from the trained SVM function,
    f(x)
    6.1   IF DCT coefficient is non-smooth THEN
          6.1.1  Extract
                 Go to Step 7
    6.2   ELSE
          6.2.1  Get DCT coefficent of 8-neighbors' LSB & Counter=+1
          6.2.2  IF DCT coefficient is non-smooth THEN
                    Go to Step 6.1.1
                 ELSE IF Counter<=8 THEN
                    Get next DCT coefficent of 8-neighbors' LSB
                       Go to Step 6.2.2
                    ELSE
                       Find DCT coefficient with highest embedding
                       strength
                       Go to Step 6.1.1
7.  IF secret-bit ≠ EOF
        Go to Step 5
    ELSE
        Go to Step 8
8.  Decrypt and decompress to obtain an original secret-bits
9.  Convert secret-bits into an original secret-message
```

From the above StegaSVM-Shifted LSB extracting algorithm steps, it can be summarized as follows that each stego-image will be divided to 8x8 non-overlapping blocks. Then, the stego-image will be transformed to DCT domain and finally quantized. After that, the process to retrieve embedding location will start on a transformed stego-image using the same key during the embedding process. After the real location has been identified, then the DCT coefficient process to determine whether it is smooth or non-smooth area can be done. This determination process is done through StegaSVM classification process utilizing the trained StegaSVM function *f*(x).

If it is found out that theDCT coefficient is made of non-smooth area then the secret-bit will be extracted. If not, get the embedding strength at the current location and then followed by getting the DCT coefficient from the 8-neighbors to the original embedding location. In each of the embedding location from the 8-neighbors, the same process will be repeated, that is, check-up on types of DCT coefficient (i.e. smooth or non-smooth areas) and also, to determine the embedding strength. If the first embedding location from the 8-neighbors location is a non-smooth area, then the secret-bit will be extracted hence, move to the next embedding location where the same process will occur. Therefore, every time the extraction process happens, the embedding location sequence file will be checked whether the status is end of file (EOF) or not. If the file status is EOF then the process will continue to decrypt and decompress to obtain original secret-bits. These secret-bits will be converted into an original secret-message. If not, then the extraction process will reoccur for the next embedding location.

IV. DISCUSSION

The development technique is comparing two types of LSB techniques, that is, Shifted LSB and Direct LSB Substitution. Shifted LSB is applied with the usage of StegaSVM-Shifted LSB while Direct LSB Substitution technique is represented by using another tool that is StegaMage that shows it in Table 1 as follows.

Table 1
PSNR of different model of LSB substitution

Types of cover-images	PSNR	
	Shifted LSB (StegaSVM-Shifted LSB)	Direct LSB Substitution (StegaMage)
Lena	49.86	32.76
Baboon	49.33	31.45
Uitm	48.89	29.01
Clock	47.68	28.91

From Table 1, it showed the usage of Shifted LSB technique resulted in having much higher PSNR for all types of cover-image. The value for all PSNR for Shifted LSB technique reached 40 and above compared to another technique. The highest PSNR is 49.86 through Lena cover-image and the lowest is 47.68 for Clock cover-image. Meanwhile, for Direct LSB Substitution technique, the highest PSNR value is only at 32.76 and the lowest is at 28.91 for Clock cover-image.

LSB Shifted technique shows that, it is not only succeeded in embedding randomizing implementation, but it even make sure that the embedding location is secured compared to direct LSB substitution.

V. CONCLUSION

There are two important types of algorithms that have been proposed in this research, that are, Shifted LSB embedding and Shifted LSB extracting algorithms. In Shifted LSB embedding algorithm, secret-message will be

encrypted and compressed to add more security or safety and also for smaller files. Secret-message that have been transformed to secret-bits which embbeded into the middlefrequency in transformed DCT cover-image, the most stableand similar magnitude. As for Shifted LSB extracting algorithm, stego-image that is send through public channel (e.g email) will firstly transformed to DCT before it is quantized. Based on the same secret key that is used during the embedding process, a sequence of location extracting will be determine and the secret-bit will be extracted. Just before the secret-bits are extracted, function *f*(x) will be executed to complete the StegaSVM-Shifted LSB technique operation, that is, to extract the right secret-message towards the end.

ACKNOWLEDGMENT

This research is funded by the Ministry of Higher Education Malaysia under the Research Acculturation Collaborative Effort (RACE) Grant Scheme Vot 1513 under UTHM grant.

REFERENCES

[1] T. Moerland. "*Leiden Institute of Advanced Computing Science",*.Retrieved January 23, 2009, from Steganography and Steganalysis: http://www.liacs.nl/home/tmoerl/privtech.pdf. 2009.

[2] M. T. Siponen and H. Oinas-Kukkonen, "A Review of Information Security Issues and Respective Research Contributions". *Database for Advances in Information Systems* , pp. 60-80. 2007

[3] K. Rabah. "Steganography-The Art of Hiding Data". *Information Technology Journal* , pp. 245-269. 2003.

[4] S. Lau, "*SANS Security Essentials GSEC*", Retrieved May 13, 2010, from An Analysis of Terrorist Groups' Potential Use of Electronic Steganography. 2003

[5] A. Cheddad, J. Condell, K. Curran, and P. M. Kevitt. " Review: Digital Image Steganography: Survey and Analysis of Current Methods". *Journal Signal Processing*, pp. 727-752. 2010.

[6] P. M. Santi, M. K. Undu. "Genetic Algorithms for Optimality of Data Hiding in Digital Images". *Bio-Inspired Information Hiding*, pp.361-373. 2009.

[7] M. A. Younes and A. Jantan. " A New Steganography Approach for Image Encryption Exchange by Using the Least Significant Bit Insertion". *International Journal of Computer Science and Network Security* , pp. 247-254. 2008.

[8] N. Jiang. "A Novel Analysis Method of Information Hiding". *International Congress on Image and Signal Processing*, pp. 621-625. 2009.

[9] L. Bin, H. Junhui, H. Jiwu, and Q. S. Yun. " A Survey on Image Steganography and Steganalysis". *Journal of Information Hiding and Multimedia Signal Processing* , pp. 142-172. 2011.

[10] J. He, S. Tan, & T. Wu. "On the Security of Steganographic Techniques". *International Congress on Image and Signal Processing*, pp. 716-719. 2008.

[11] R. Din and A. Shamsudin." Digital Steganalysis: Computational Intelligence Approach". *International Journal of Computers* , pp. 161-16. 2009.

[12] S. Trivedi and R. Chandramouli. " Active Steganalysis of Sequential Steganography". *SPIE Conference*, 2003 pp. 123-130.

[13] C. Cachin. " An Information-Theoretic Model for Steganography". *Information and Computation* ,2004, pp. 41-56.

[14] Y. K. Lee and L. H. Chen. "A Secure Robust Image Steganographic Model". *The Conference on Information Security*, 2008, pp. 275-284. Hualien, Taiwan

[15] T. Morkel, J. H. Eloff, J. and M.S. Olivier, M. "An Overview of Image Steganography". *Annual Information Security South Africa Conference.* 2005.

[16] Z. Duric, D. Richards, and Y. Kim. "Minimizing the Statistical Impact of LSB Steganography". *International Conference on Image Analysis and Recognition*, 2005 pp. 1175–1183). Toronto: Springer

[17] S. Ramly, S.A. Aljunid and H.S. Hussain. "SVM-SS Watermarking Model for Medical Images".*Digital Enterprise and Information Systems*, 2011, pp. 372-386.

[18] J. Fridrich. "Featured-Based Steganalysis for JPEG Images and Its Implications for Future Design of Steganographic Schemes". *International Workshop on Information Hiding*, 2006, pp. 67-81.

[19] R. W. Lucky, "Automatic equalization for digital communication," *Bell Syst. Tech. J.*, vol. 44, no. 4, pp. 547–588, Apr. 1965.

[20] S. P. Bingulac, "On the compatibility of adaptive controllers (Published Conference Proceedings style)," in *Proc. 4th Annu. Allerton Conf. Circuits and Systems Theory*, New York, 1994, pp. 8–16.

[21] A. Almohammad, R. M. Hierons, and G. Ghinea. " High Capacity Steganographic Method Based Upon JPEG". *The Third International Conference on Availability, Reliability and Security* , 2008, pp. 544-549.

[22] M. A. Zaher. "Modified Least Significant Bit (MLSB)". *Computer and Information Science* , 2011, pp.60-67.

[23] E. Walia, P. Jain, and P. Navdeep. " An Analysis of LSB and DCT Based Steganography". *Global Journal of Computer Science and Technology* , 2010 pp.4-8.

[24] L. Por, W. K Lai, Z. Alireza, T. F. Ang, M. Su, B. Delina. "StegCure: A Comprehensive Steganographic Tool using Enhanced LSB". *WSEAS Transactions of Computers Journal* , 2008 pp. 1309-1318.

[25] V. Kecman. *Learning and Soft Computing*. London: The MIT Press. 2001

[26] L. Li, W.Y. Ding, and J.Y Li. "A Novel Robustness Image Watermarking Scheme Based on Fuzzy Support Vector Machine". *The 3rd IEEE International Conference on Computer Science and Information Technology*, 2010, pp. 533 – 537.

[27] F. Meng, H. Peng, Z. Pei, J. Wang, J." A Novel Blind Image Watermarking Scheme Based on Support Vector Machine in DCT Domain". *International Conference on Computational Intelligence and Security*, 2008, pp. 16 - 20.

[28] H. H. Tsai, H.C. Tseng, Y.S. Lai." Robust Lossless Image Watermarking Based on A-Trimmed Mean Algorithm and Support Vector Machine". *The Journal of Systems and Software*, 2008 pp. 1015–1028.

[29] H. Peng, J. Wang, W. Wang. "Image Watermarking Method in Multiwavelet Domain Based on Support Vector Machines". *The Journal of Systems and Software*, 2010, pp. 1470–1477.

[30] C. Schneider, W. Rasband, and K. Eliceiri. " NIH Image to ImageJ: 25 years of image analysis. *Nature Methods*, 2012, pp. 671–675