Performance of Efficient Steganographic Methods for Image and Text

Linqiang Ouyang, Jin H. Park, and Harbhinder Kaur
Computer Science Department, California State University, Fresno, CA 93740, U.S.A.
Email: jimmyou587@gmail.com, jpark@csufresno.edu, harbhinder_kaur20@mail.fresnostate.edu

***Abstract*—Image steganography is a popularly used method of conducting secure on-line communication. We developed and tested four different models, which are based on the secret key enhanced LSB (Least Significant Bit) based approach, of hiding image and/or text for performance in the aspects of data type and size, processing time, and the quality of the stego-image. Those four models include image only, text only, and two proposed hybrid models for hiding image and text together. Our experimental results show a comprehensive performance metric of considering various aspects in the image steganography. It is also demonstrated that our proposed model (optimized version) of hiding image and text together outperforms the straight-forward implementation of the hybrid model in terms of processing time and capacity.**

***Index Terms*—image steganography, cover image, stego-image, bit stream, security**

I. Introduction

As Internet is so widely used nowadays, security issues become more serious on the on-line communication. Steganography is one of the techniques aiming to achieve the secure communication between authorized parties. Different from the cryptography, which encodes secure information into unreadable format, steganography hides secure information within a medium to establish an invisible communication [1], [2]. Image Steganography is a popularly used approach to hide information, including image, text, video, etc., into a cover image without noticeable distortion so that hidden information is not visible to human eyes. An ideal Image Steganography method should also make the hidden information hard to be detected electronically. There have appeared diversified researches on the image steganography in the literature including the methodologies of hiding grey-scale/color images and text [3]-[10]. A relatively simple and effective method of hiding information in the cover image is storing the information into specific positions, i.e., least significant bit (LSB) in each pixel, of the cover image, but the level of the security is pretty low, i.e., it is easy to detect and extract the information from the stego-image. Some later works [3]-[6] focused on adding enforced quality and security to the LSB based methodology.

Manuscript received July 20, 2015; revised October 11, 2015.

Although there appeared many approaches in the image steganography, there has not appeared any comprehensive study on performance comparison in various aspects. In fact, different researches have been conducted under different environments and assumptions and this makes it difficult to do the cross-approach comparison. Even in each research work, it is not shown clearly the maximally allowed size of the information, which depends on the specific method used, to hide and the relationship between performance and data size used. This becomes the motivation of our research.

In this paper, we provide a comprehensive performance measurement on several different models of hiding image and/or text information in the image steganography. We built four models to test; two are based on currently available security enhanced LSB based approaches described in [6] and [8] for image and text, respectively, and we developed a couple of proposed hybrid models to hide both image and text together with the secret key mechanism. The first hybrid model is a straightforward implementation of manipulating the image and text information combined, and the second hybrid model uses an optimized method of manipulating the image and text bit streams simultaneously to achieve gains in time, quality and capacity. Thus, our efforts are in two fold, providing an efficient hybrid model of manipulating both image and text together and providing a comprehensive performance metric showing the relationship among information type and size, processing time and quality.

The rest of this paper is organized as follows. In Section II, a brief review on the related work is presented. In Section III, four models of the image steganography including our proposed hybrid models of manipulating image and text together are described. In Section IV, experimental results and performance analysis are presented. Finally, Section V concludes the paper.

II. Related Work

In this section, we briefly review some LSB based image steganography approaches, which became the backbone of our proposed and tested models.

To recover the drawback of the naïve LSB based approach both non-adaptive and adaptive approaches were proposed. The work described in [3] proposed a non-adaptive method of improving the quality of the stego-image by using the optimal pixel adjustment

doi: 10.12720/jait.7.1.29-33

process. A more effective approach, which is an adaptive method based on the inter-pixel relationship, is described in [4]. A further effective method of improving the quality of the stego-image by utilizing each pixel's dependency on neighborhood pixels is described in [5]. In this work, three different methods named four-neighbors, eight-neighbors and diagonal-neighbors are developed and tested in the gray-scale image environment. A recent work described in [6] firstly used a secret key based method and achieved higher performance in the quality of the stego-image, as well as the higher security during the online communication. In this approach, 24-bit RGB color image is used and a secret key decides the positions of hidden information being hided in the cover image. Each character in the secret key is converted to 8-bit binary values of its ASCII code and concatenated into a 1-dimensional circular bit stream. Followed by this secret key based approach, a methodology of hiding text information under the cover image using the secret key is proposed in [8].

III. Four Models of Hiding Image and Text

To yield the comprehensive performance metric of hiding image and/or text in image steganography we implemented four models based on currently available methodologies for separate processing of image and text and our own proposed methodologies for processing image and text together. All of these models use secret key based methods and we use 24-bit RGB color images and 8-bit ASCII characters in our practice.

A. Model 1 – Image Only

The first model we implemented is for hiding an image in the cover image, and the implementation is based on the secret key based LSB approach described in [6]. In this model, each character in the secret key is converted to 8- bit binary values of its ASCII code and concatenated into a 1-dimensional circular bit stream. All RGB values (0~255) of both cover and hiding images are converted to 8-bits binary values. The binary bits of the hiding image are combined into a 1-D bit stream. We summarize the hiding and extraction operations as shown below.

Hiding process (by sender):
Step 1. Get LSB of Red from cover image and get 1 bit from secret key bit stream;
Step 2. If XOR of Step1 values = 1, replace LSB of Green by 1 bit of hiding image; else, replace LSB of Blue by 1 bit of hiding image;
Step 3. If not the end of hiding data, go to Step 1.
Extraction process (by receiver):
Step 1. Get LSB of Red from cover image and get 1 bit from secret key bit stream;
Step 2. If XOR of Step1 values = 1, pick LSB of Green matrix; else, pick LSB of Blue matrix;
Step 3. Store the bit from Step 2 into 1-D array;
Step 4. If not the end of hiding data, go to Step 1.

B. Model 2 – Text Only

The second model is for hiding text in the cover image, which can be done with the analogous process used in Model 1. The only difference from Model 1 is that each character in the hiding text is converted to an ASCII code and the corresponding bit stream is generated. A secret key based text hiding methodology is described in [8] and we skip describing the details of the method in this paper.

C. Model 3 – Image+Text (Sequential)

To hide image and text together within a cover image, we propose and test a couple of methodologies, one is based on the sequential straightforward mechanism (Model 3) and the other is an optimized approach (Model 4).

In Model 3, operations used are a combination of the operations used in Model 1 and Model 2, i.e., hiding and extracting image and text sequentially. Most steps of hiding and extracting processes are analogous to the ones described in Model 1, except some minor changes described as follows. In the hiding process, between Step1 and Step 2, we need to concatenate hiding image and text bit streams together as one hiding bit stream. In Step 3 of the extraction process, we need to store the bits into hiding image and text bit streams separately.

To extract image and text appropriately, the mode and sizes of hidden image and text should be known beforehand along with the secret key. As used in other approaches, we store these information at the beginning of the stego-image.

D. Model 4 – Image+Text (Optimized)

With the third model, it is obvious that we cannot hide information (image+text) bit stream that is longer than the total number of pixels in the cover image. This is due to the technology that only one RGB value is used in each pixel of the cover image – waste of resources. This becomes the motivation of developing our optimized version, Model 4. In fact, it is observed that when the LSB of Green or Blue value is substituted, the one left in the same pixel, either Blue or Green, is still unchanged.

In Model 4, each bit from hiding image or text bit stream is hided in the Green or Blue value of the same pixel and this increases the hiding capacity in the cover image. In other words, the LSBs of both Green and Blue values in each pixel are replaced until at least one of the hiding image and text bit streams is ended. During the hiding process, if the XOR value is 1, then the LSB of Green value is overwritten by one bit from the hiding image bit stream and at meanwhile, the LSB of Blue value is replaced by one bit from hiding text bit stream. If the XOR value is 0, the other path is taken. This approach doubles the space in the cover image to hide image and text, but still keeps the stego-image with no significant distortion. Fig. 1 illustrates the operations of the hiding process used in Model 4.

In order to extract the hidden image and text, it is need to know the sizes of hidden image and text, as well as the secret key. The extraction process is illustrated in Fig. 2. As shown in the figure, if the XOR value is 1, the LSB of Green value is retrieved and stored into the hidden image bit stream. The LSB of Blue value is added into the hidden text bit stream. If the XOR value is 0, the LSB of

Blue value joins the hidden image bit stream. The LSB of Green value is appended to the hidden text bit stream.

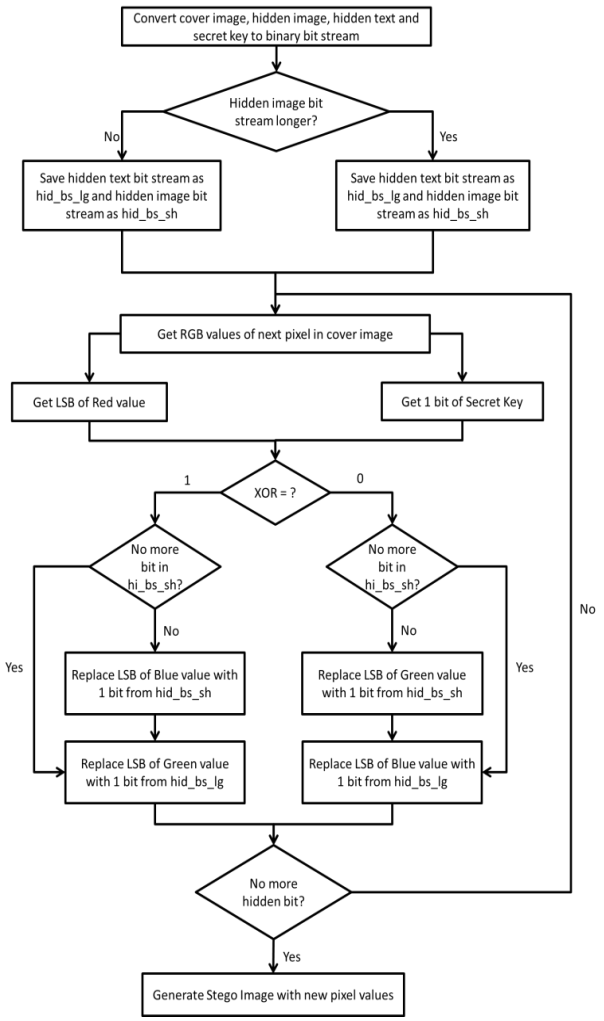

Figure 1. Hiding process of Model 4

IV. Experimental Results

We implemented the four models described in Section III in Python and executed them on a Mac machine with i5 processor (2.7 GHz) and 8 GB RAM. In our practice, we used maximum sized hidden image and/or text in each model and tested with varying sizes of the data. The cover image and the hidden image used are Lenna picture (RGB, 512x512 pixels) and Mercedes picture (RGB, 127x86 pixels), respectively. These images are shown in Fig. 3. In Model 1 and 3, the Mercedes picture is the biggest picture that can be hidden in the cover image. In Model 2, the longest text that the cover image (Lenna) can hold is 32,768 bytes long. Although Model 4 can hold more hidden data than Model 3, as described in Section III-D, we used the same sized data for the purpose of performance comparison. Performances are measured in terms of processing time and the quality of the stego-image, which holds the hidden information. To measure the relationship between the performance and hidden information size, we used 10 different sized hidden information in each model, i.e., 10%~100% of the original maximum sized data. The quality of the stego-image is determined by the PSNR (Peak Signal-to-Noise Ratio) value, as regularly used in other research, and the formulas are shown below in which images I and K both have m and n as the width and height, respectively, and the Max value of I is 255 in the true color image.

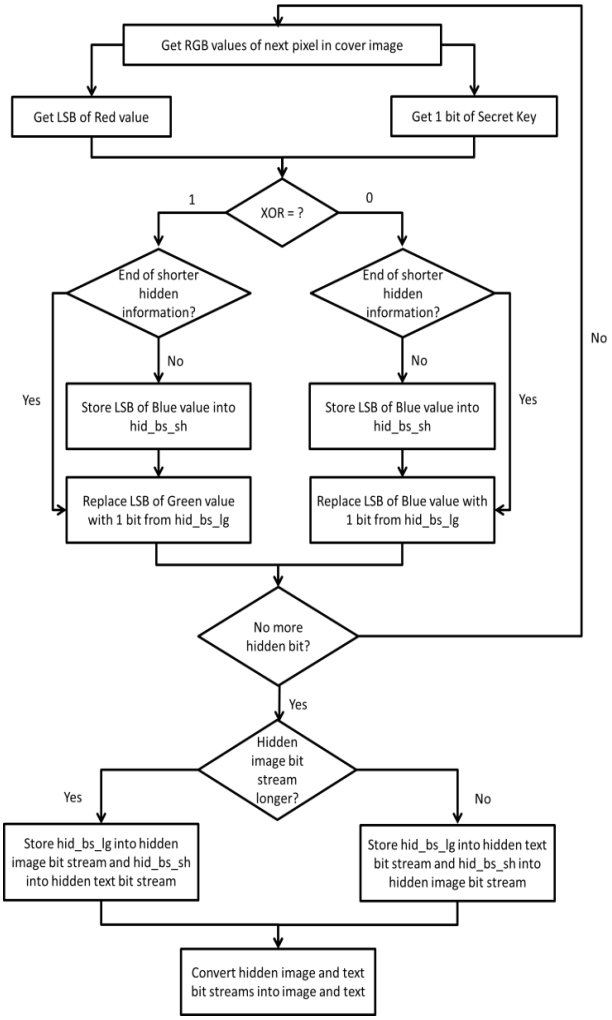

Figure 2. Extraction process of Model 4

Figure 3. Cover image and hidden image used

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2 \quad (1)$$

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) \quad (2)$$

It is obvious that the higher PSNR value represents the better quality of the stego-image and thus, more secure from unauthorized users' attack.

Table I shows the performance of Model 1 (image only) tested with different sized hidden images. As shown in the table, the processing time increases linearly with the increasing sized hidden images, while the stego-image quality decreases in the similar pattern.

Table II shows the performance of Model 2 (text only) tested with different sized hidden texts. As shown in the table, processing time and stego-image quality change in a linear pattern depending on the text size, but the stego-image qualities are much higher than the ones shown in Model 1. This means that hiding ASCII text in the cover image has higher stego-image quality than hiding RGB image for both maximum and reduced sizes.

TABLE I. PERFORMANCE OF MODEL 1

Hidden Image (Mercedes) Size	Processing Time (sec)	PSNR (in dB)
13 x 86(10%)	0.458	65.8141
25 x 86(20%)	0.532	62.9893
38 x 86(30%)	0.635	61.1458
51 x 86(40%)	0.715	59.8704
64 x 86(50%)	0.798	58.8826
76 x 86(60%)	0.912	58.1427
89 x 86(70%)	0.997	57.4413
102 x 86(80%)	1.127	56.8593
114 x 86(90%)	1.222	56.3785
127 x 86(100%)	1.362	55.9046

TABLE II. PERFORMANCE OF MODEL 2

Hidden Text Size (byte)	Processing Time (sec)	PSNR (in dB)
3276(10%)	0.498	82.9673
6553(20%)	0.526	79.9634
9830(30%)	0.602	78.2120
13107(40%)	0.739	76.9791
16384(50%)	0.861	76.0136
19660(60%)	0.923	75.2227
22937(70%)	1.015	74.5355
26214(80%)	1.134	73.9407
29491(90%)	1.244	73.4119
32768(100%)	1.341	72.9727

To measure the performances of Model 3 and Model 4, we set the size of the hidden image with the 50% one (64x86) used in testing Model 1 and tested with varying sized texts accommodated in the remaining parts of the cover image. With the 50% sized hidden image, we could maximally hide 16,256 bytes (100%) of text in Model 3, and we use same sized image and text in testing Model 4 for the sake of fair comparison though Model 4 can hide much more data than Model 3. In the case with the 64x86 (50%) hidden image, Model 4 can hide maximally 32,768 bytes of text with the processing time of 1.754 sec and PSNR value 54.1151. Table III and Table IV show the performances of Model 3 and Model 4, respectively.

TABLE III. PERFORMANCE OF MODEL 3

Hidden Image Size (fixed)	Hidden Text Size (byte)	Processing Time (sec)	PSNR (in dB)
64 x 86	1625(10%)	0.875	58.4778
64 x 86	3251(20%)	0.887	58.0955
64 x 86	4876(30%)	0.898	57.7509
64 x 86	6502(40%)	0.958	57.4303
64 x 86	8128(50%)	1.082	57.1333
64 x 86	9753(60%)	1.033	56.8546
64 x 86	11379(70%)	1.126	56.5947
64 x 86	13004(80%)	1.202	56.3475
64 x 86	14630(90%)	1.212	56.1155
64 x 86	16256(100%)	1.298	55.8937

TABLE IV. PERFORMANCE OF MODEL 4

Hidden Image Size (fixed)	Hidden Text Size (byte)	Processing Time (sec)	PSNR (in dB)
64 x 86	1625(10%)	0.824	58.4662
64 x 86	3251(20%)	0.852	58.0921
64 x 86	4876(30%)	0.908	57.7452
64 x 86	6502(40%)	0.872	57.4220
64 x 86	8128(50%)	0.856	57.1253
64 x 86	9753(60%)	0.891	56.8456
64 x 86	11379(70%)	0.878	56.5831
64 x 86	13004(80%)	0.948	56.3363
64 x 86	14630(90%)	0.937	56.1014
64 x 86	16256(100%)	0.970	55.8802

As shown in Table III and Table IV, we could observe that the stego-image qualities are pretty analogous in the two models with the given identical data, but the processing time of Model 3 is more rapidly increasing with the given increasing sizes than that of Model 4. Although it is not shown in Table IV, Model 4 can hide a lot more data than Model 3, as we mentioned earlier. Thus the processing time efficiency and the capacity of hiding data in Model 4 are higher than Model 3.

V. Conclusions

We implemented and compared four different models of hiding image and/or text in the cover image. Two models are built based on the currently available secret key based LSB methodologies for hiding image and text separately. Other two models are our proposed hybrid approaches of hiding image and text together in the cover image. In our practice, we used varying sized image and text to yield a comprehensive metric showing the relationship between the performance and hiding information type and size. Our experimental results also show that the optimized version of the proposed approach of hiding image and text together achieves higher performance than the straightforward version in both processing time and capacity.

Our future study includes developing more diversified models and testing more diversified data to generate more comprehensive metric of the performance.

References

[1] F. Hartung and M. Kutte "Information hiding-a survey," in *Proc. of the IEEE*, vol. 87, no. 7, pp. I062-I078, July 1999.

[2] A. Westfeld and G. Wolf, "Steganography in a video conferencing system," in *Proc. the 2nd International Workshop on Information Hiding*, Springer, vol. 1525, 1998, pp. 32-47.

[3] C. K. Chan and L. M. Cheng, "Hiding data in images by simple LSB substitution," *Pattern Recognition*, vol. 37, pp. 469–474, March 2004.

[4] D. C. Wu and W. H. Tsai, "A steganographic method for images by pixel-value differencing," *Pattern Recognition Letters*, vol. 24, no. 9, pp. 1613-1626, 2003.

[5] M. Hossain, S. A. Haque, and F. Sharmin, "Variable rate steganography in gray scale digital images using neighborhood pixel information," in *Proc. the 12th International Conference on Computer and Information Technology*, Dhaka, Bangladesh, Dec. 2009, pp. 21-23.

[6] S. M. M. Karim, Md. S. Rahman, and Md. I. Hossain, "A new approach for LSB based image steganography using secret key," in *Proc. 14th International Conference on Computer and Information Technology*, Dhaka, Bangladesh, Dec. 2011, pp. 22-24.

[7] N. Jain, S. Meshram, and S. Dubey, "Image steganography using LSB and edge-detection technique," *International Journal of Soft Computing and Engineering*, vol. 2, no. 3, July 2012.

[8] D. Rawat and V. Bhandari, "Steganography technique for hiding text information in color image using improved LSB method," *International Journal of Computer Applications*, vol. 67, no. 1, pp. 22-25, April 2013.

[9] W. Q. Luo, F. J. Huang, and J. W. Huang, "Edge adaptive image steganography based on LSB matching revisited," *IEEE Trans. on Information Forensics and Security*, vol. 5, no. 2, pp. 201-214, June 2010.

[10] D. Singla and M. Juneja, "An analysis of edge based image steganography techniques in spatial domain," in *Proc. 2014 International Conference on Recent Advances in Engineering and Computational Sciences*, Chandigarh, March 2014, pp. 1-5.

Linqiang Ouyang is a graduate student in the Department of Computer Science at California State University, Fresno, CA USA. He received his B.E. degree in Electronics Science and Technology from Nanjing University of Posts and Telecomm-unications, China in 2007. His research interests include network security, natural language processing, and machine learning.

Jin H. Park is an assistant professor in the Department of Computer Science at California State University, Fresno, CA, USA. He received his Ph.D. degrees in Computer Science from Oklahoma State University, Stillwater, OK, USA in 1998. His research interests include high performance computing, parallel and distributed processing, bioinformatics and computational biology, and network.

Harbhinder Kaur is a graduate student in the Department of Computer Science at California State University, Fresno, CA USA. Her research interests include cryptography and network.