



Steganography of Halftone Images by Group Alteration of Grayscale Pixels

Mojtaba Mahdavi^{a,*} Shadrokh Samavi^b

^aFaculty of Computer Engineering, University of Isfahan, Iran.

^bDepartment of Electrical and Computer Engineering, Isfahan University of Technology, Iran.

ARTICLE INFO.

Article history:

Received: 13 November 2014

Revised: 09 December 2015

Accepted: 10 January 2016

Published Online: 10 February 2016

Keywords:

Steganography, Image Halftoning, Blind Steganalysis

ABSTRACT

Data hiding in halftone images is a challenging task since these images are very susceptible to alterations. Current steganographic methods for these types of images either embed the data after the image is completely halftoned or modify the halftoning process for embedding purposes. In this paper we present a third type of steganography which performs the embedding by altering the grayscale image prior to its conversion to the halftone form. The goal of the proposed algorithm is to minimize the amount of alterations that are caused in the grayscale image. We compared our algorithm with a secure steganographic method. Also an effective steganalytic attack is applied to show the security of the proposed algorithm and the secure capacity of the proposed method was calculated to be 2% of bi-level image size while the secure capacity of its counterpart was calculated to be under 1%. It is shown that the proposed algorithm in terms of visual quality and security, as compared to the existing algorithms, produces superior results.

© 2015 JComSec. All rights reserved.

1 Introduction

Steganography is a branch of the science of information hiding where other branches include cryptography and watermarking. In steganography the existence of the communication has to be kept secret and the goal is to mask the very presence of communication, making the true message not discernible to the observer [1]. Cover medium is where the secret message is to be embedded in. The cover medium becomes a stego medium after a secret message is embedded in it. A popular cover medium is image. Distinguishing between cover and stego images is the responsibility of steganalysis techniques [2]. This distinction is performed without

knowing the embedding key and sometimes without knowing the steganographic technique.

Halftone images are used in many applications, such as lithographic printing of books and magazines as well as computer printers. Halftoning is the process of converting a grayscale image to an image where the pixels can have 0 or 255 (black or white) intensities. A widely used halftoning scheme is that of Floyd-Steinberg (FS) [3]. This algorithm is simple and produces high quality halftone images from the grayscale pictures.

In the FS method, the intensity of a grayscale pixel is compared with a threshold of $255/2$. If the pixels value is bigger than the threshold its halftone value becomes 255, otherwise, the new value of the pixel becomes 0. Hence, each halftone pixel can have one of the two values and can be defined with only one bit. This changing of the value of a pixel produces an error.

* Corresponding author.

Email addresses: m.mahdavi@eng.ui.ac.ir (M. mahdavi)
samavi96@cc.iut.ac.ir (S. Samavi)

ISSN: 2322-4460 © 2015 JComSec. All rights reserved.



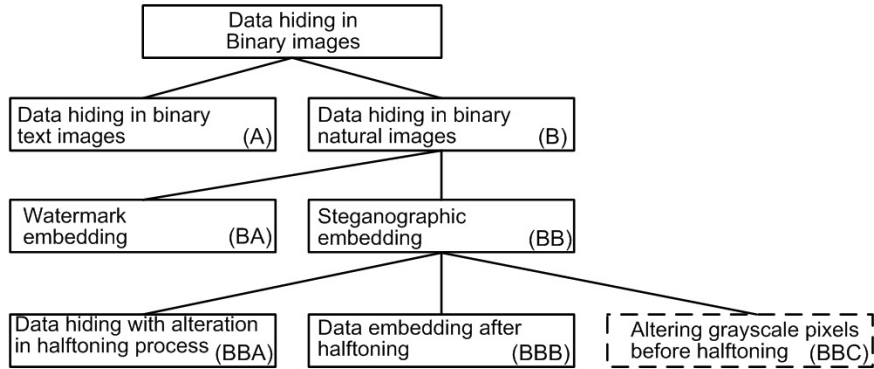


Figure 1. Different categories of data hiding methods in binary images

Suppose the original grayscale value of a pixel is I and its halftone value is 255. Then the error value would be $e = I - 255$. The FS algorithm diffuses this error into the neighboring pixels [3]. The error from the halftoning of the pixel is diffused into the neighboring pixels by adding a fraction of the error to each one of neighbors. The FS algorithm starts with the top left corner of the image and goes toward right in a row by row manner. Therefore, halftoning of a pixel has no effect on the pixels which have previously been halftoned.

Besides FS, there are other halftoning algorithms [4]. In this paper we have used the FS algorithm for halftoning. This is due to the fact that FS is simple and produces good quality images. Other halftoning algorithms can be used as a part of the proposed method.

Different embedding schemes have been proposed for halftone images which some of them are reviewed in this paper. The effects that embedding of data creates in an image is similar to the addition of noise to that image. Unlike grayscale images, halftone images are very susceptible to noise addition (data embedding) which makes them vulnerable to attacks [5, 6]. The reason is that in halftone images low embedding rates under one percent are reliably detected. In this paper we are proposing a new scheme for this purpose that embeds data with minimum changes in a halftone image. The data is first compressed and it is then encrypted. This gives a noise-like characteristic to the data that is to be embedded. The proposed algorithm embeds the desired data in the grayscale image prior to the halftoning process. This is done in such a manner that the secret data can be extracted from the halftone image. In order to keep the visual quality of the embedded image, the proposed algorithm considers the complexity of a group of pixels before embedding data in that group. By doing so, the algorithm avoids embedding in regions that potentially may reveal the presence of the hidden data. The results of the

proposed method are compared with a powerful comparable algorithm [7]. To have a fair comparison, we used the most effective available steganalytic attack on both methods. The proposed method is proved to be far more secure than the other methods.

The paper is organized such that a number of data hiding algorithms for halftone images are reviewed in Section 2. Also in Section 2 some of the existing steganalysis methods are reviewed. The proposed Soft-Steg algorithm is detailed in Section 3 and the implementation results are presented in Section 4 of the paper. Section 5 compares the security of proposed method with a reference algorithm. The paper is concluded in Section 6.

2 Existing Data Hiding Algorithms

To show the position of our proposed algorithm with respect to the existing schemes we have categorized data hiding algorithms in binary images according to the diagram of Figure 1.

Generally, these algorithms try to modify edges of characters for embedding purposes. Examples of such algorithms are presented in [8, 9]. The second major group (type B) contains data hiding schemes that try to embed data into natural binary images. An example of such image is Figure 3a. Embedding in such images can be categorized as two subcategories. The first category (type BA) belongs to algorithms that embed an image (such as a logo or a copyright mark) inside a target image. Since these algorithms are used in watermarking, the exact extraction of the embedded data is not the main concern of such schemes. Examples of such algorithms can be found in [10–17]. On the other hand, there are algorithms that are intended for steganography in binary images where the embedded data has a general nature (type BB). Steganographic schemes perform the embedding either by altering the halftoning process [7, 18] (type



BBA) or by performing the embedding after the image is halftoned [18] (type BBB). We propose a new group of algorithms as those that alter the grayscale pixels prior to the halftoning of the image without any change in the halftoning process (type BBC). To the best of our knowledge, the proposed algorithm is the only method which falls into this last category.

Before getting into the details of the proposed method, some of the existing data hiding algorithms that are used for halftone images are reviewed. Fu et al. [18] proposed two data hiding methods, which perform embedding by modifying the halftoning process (type BBA). These two methods are called the data hiding error diffusion (DHED) and the modified data hiding error diffusion (MDHED). Both DHED and MDHED start embedding in a location by forcing the pixel to be either 0 or 255 and they use error diffusion in the neighboring pixels. In DHED the error is diffused in pixels which do not have any embedded data yet. Fu et al. try to remedy the created artifact by offering the modified version of the algorithm (MDHED) which diffuses the error both in previous as well as in the upcoming pixels.

Pei et al. [14] proposes a watermarking scheme (type BA) which is performed during the error diffusion process. This algorithm is suitable for watermarking but since the original image is required for the extraction purposes, the algorithm cannot be used for steganography. Also the watermark can be extracted with some loss of data which is not tolerable in steganography. Another BA type algorithm is proposed by Yip et al. [11]. In that method, $N \times N$ blocks of the image are classified based on the distribution of black and white pixels in the block. To maintain the intensity of the image only blocks that have equal number of zeros and ones are selected for embedding. Depending on the data that is being embedded, all of the pixels of the block are complemented or the block stays unchanged.

A number of steganalytic attacks are also available that are able to detect the presence of the embedded data in binary text images. Examples of such techniques are presented in [19–23]. In one of the most recent steganalysis techniques, Cheng et al. [5] proposed an inverse halftoning based steganalysis method of halftone images. Inspired by the steganalysis techniques in [24] for gray level images and the fact that a halftone image is obtained through a halftone process on a gray level image, they extended the scheme in [23] to a general method for the steganalysis of halftone images. In this method the wavelet statistic features are extracted from the reconstructed gray level image through the inverse halftoning of a given image. These features are then used in a support vector machine (SVM) classifier.

Among various steganalytic attacks, the method proposed by Cheng et al. [5] is the most accurate one. We will use this attack to show the security of our steganographic algorithm. In [5] securities of many embedding algorithms are analyzed. Some improved versions of Fus algorithms, which are presented in [25], along with DHED and MDHED are analyzed in [5]. Cheng showed that the security levels of DHED and MDHED (type BBA algorithms) are better than the other BBB type algorithms proposed by Fu et al. Furthermore, Cheng has showed that the security levels of DHED and MDHED are almost the same. Therefore, to show the security of our proposed method, we will compare it with DHED.

3 Proposed Method

In this section we propose a steganographic method called SoftSteg, which embeds data in natural binary images. To reduce visual artifacts of the stego image, the algorithm avoids embedding in non-complex regions.

To better explain the proposed method, we use the notations depicted in Table 1.

The embedding function of the proposed method consists of the steps in Algorithm 1.

Table 1. Notations used in this paper

Notation	Description
I	The original gray scale image
$n \times m$	Number of rows (n) and columns (m) of the grayscale image
X_k	Each pixel element of image (I) such that $X_k \in \{0, \dots, 255\}$ and $k \in \{0, \dots, n \times m - 1\}$
N	The number of bits to be embedded in each group of pixels
L	The number of pixels in a group
g_i	The i^{th} group of pixels such that $g_i = \{x_{i \times L}, \dots, x_{i \times L + L - 1}\}$ and $i \in \{0, \dots, \lfloor \frac{n \times m}{L} \rfloor\}$
T	The complexity threshold
H	The halftoning operation
$w(g_i)$	Number of white pixels in $H(g_i)$
\mathcal{C}	The set of all possible outcomes of g_i such that $ L - 2w(g_i) < T$
d_j	The j^{th} group of data bits with length N
e_i	A group of errors with L elements to be added to g_i for data embedding, $e_i = e_0, \dots, e_{L-1}$
X	The extraction function devised for a group of halftoned pixels



Algorithm 1 Embedding algorithm

```

Set  $i$  and  $j$  to zero
for  $i = 0$  to  $\lfloor \frac{n \times m}{L} \rfloor$  do
  if  $g_i \in \mathcal{C}$  then
    Find the smallest  $e_i$  such that  $d_j = X(H(g_i + e_i))$  and  $g_i + e_i \in \mathcal{C}$ 
     $g_i \leftarrow g_i + e_i$ 
    Halftone  $g_i$  to propagate the error
     $j \leftarrow j + 1$ 
  else
    Halftone  $g_i$  to propagate the error
  end if
end for

```

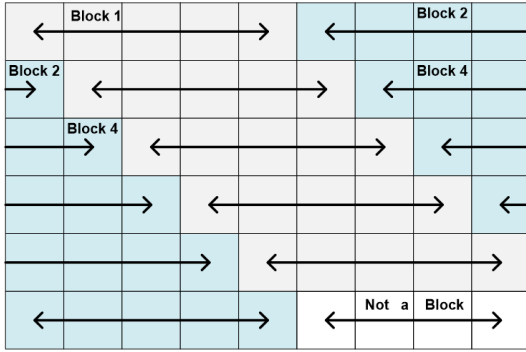


Figure 2. An example of the grouping scheme

For the embedding purpose, each group of pixels (g_i) is tested for complexity. If the embedding is performed only in complex groups of pixels then the probability of detection will be minimized and the quality of the stego image will remain high. The set of all complex groups is \mathcal{C} . A group of pixels with length L is complex if the number of its white pixels $w(g_i)$ is such that $|L - 2w(g_i)| < T$. By considering such criterion, it is guaranteed that the embedding is not done in mainly white or mainly black groups. For a complex group, some of the grayscale pixels are altered by adding a group of errors, e_i , such that when the pixels are halftoned the embedded data can be extracted from that group. The added errors should be the smallest values that perform the task. This means that the grayscale pixels are minimally altered. The procedure of finding such group of errors, is discussed later in the paper. Furthermore, the group of pixels should remain complex after the addition of the errors. Figure 2 shows an example of the grouping scheme that is used in our algorithm. Some of the groups may start on one row of the image and continue to the next row. In the shown example, the length of a group, L , is 5.

The process of extracting the data from the halftoned stego-image is as Algorithm 2:

The extraction process is performed on the halftoned stego-image. The image is partitioned

Algorithm 2 Extraction algorithm

```

Set  $i$  and  $j$  to zero
for  $i = 0$  to  $\lfloor \frac{n \times m}{L} \rfloor$  do
  if  $g_i \in \mathcal{C}$  then
     $d_j \leftarrow X(g_i)$ 
     $j \leftarrow j + 1$ 
  end if
end for

```

into groups of length L and the complex groups are detected. In the i^{th} group of pixels which is a complex one, the extracted data d_j is calculated from the number of white pixels, $w(g_i)$. In other words, $X(g_i) = w(g_i) \bmod 2^N$.

There can be various procedures for finding a small group of errors, e_i , to be added to the grayscale pixel group. One can use a full search over all possible error groups to find the smallest group of errors which is obviously a very time consuming procedure and is not applicable. Thus alternative procedures should be taken into account which may not find the global smallest group of errors for each group.

An example of such procedure consists of finding two candidate groups for e_i . These two groups of error values are called e_u and e_d . The group that has smaller set of values is chosen. Algorithm 3, shows one of the various methods that one can search for a minimal set of errors.

Algorithm 3 A minimum error searching algorithm

```

Set all elements of  $e_u$  and  $e_d$  to zero.
Set  $k \leftarrow 0$ 
while  $d_j \neq X(H(g_i + e_u))$  do
  Set  $e_{u_k} \leftarrow e_{u_k} + 1$ 
   $k \leftarrow (k + 1) \bmod L$ 
end while
Set  $k \leftarrow 0$ 
while  $d_j \neq X(H(g_i + e_d))$  do
  Set  $e_{d_k} \leftarrow e_{d_k} - 1$ 
   $k \leftarrow (k + 1) \bmod L$ 
end while
if  $\sum_{k=0}^{L-1} e_{u_k} < - \sum_{k=0}^{L-1} e_{d_k}$  then
   $e_i \leftarrow e_u$ 
else
   $e_i \leftarrow e_d$ 
end if

```

Both e_u and e_d are strings of length L which initially have all zero elements. An intended group of pixels g_i goes through two tests. First of all g_i is altered by adding e_u to it and the altered pixels are halftoned. If the desired d_j cannot be extracted from the halftoned altered pixels, then one of the elements of e_u is incremented. If this step is not successful the next element of e_u is incremented. This incrementing is performed



in a round robin fashion for the elements of e_u until d_j is extractable. Same procedure is performed for e_d except that an element of this error vector is decremented every time that the alteration process does not result in the correct extraction of d_j . Finally, e_u and e_d are compared to find out which one produces the least amount of error.

An example of explained procedure is as follows. Imagine secret data bit $d_j = 0$ is considered for embedding in a group of 5 ($L = 5$) pixels $g_i = \{125, 78, 76, 75, 73\}$ as cover pixels. Halftoned version of g_i is $H(g_i) = \{0, 255, 0, 0, 0\}$ and number of white pixels is $w(g_i) = 1$. According to above procedure e_u can be found as $e_u = \{4, 4, 4, 4, 4\}$ and e_d as $e_d = \{-4, -4, -4, -3, -3\}$. Halftoned version of $g_i + e_u$ and $g_i + e_d$ is $H(g_i + e_u) = \{255, 0, 0, 0, 255\}$ and $H(g_i + e_d) = \{0, 0, 0, 0, 0\}$ respectively. Number of white pixels of $g_i + e_u$ is $w(g_i + e_u) = 2$ and number of white pixels of $g_i + e_d$ is $w(g_i + e_d) = 0$. In both cases, the extracted data bit will be 0. Because $\sum_{k=0}^{L-1} e_{u_k} > -\sum_{k=0}^{L-1} e_{d_k}$ then e_d will be selected and g_i will be replaced with $\{125, 78, 76, 75, 73\} + \{-4, -4, -4, -3, -3\} = \{121, 74, 72, 72, 70\}$.

The idea behind this method of searching for a small group of errors is to find a small value that changes the number of white pixels in a group. Obviously, because e_u has all positive or zero values and it is added to the pixels of the group, the overall intensity of the group will be increased and therefore the number of white pixels will be increased too. On the other hand, since the values of e_d is negative and is added to the pixel values of the group, the overall intensity of the pixels will be decreased and this will cause the number of white pixels after halftoning to be decreased. So, intuitively it is obvious that this procedure converges to find a small group of errors to be added or subtracted from the pixel values such that the number of white pixels in half-toned version of the group is changed. To find e_i , we used the above procedure, but other methods, such as full search, can also be applied which is very time consuming. To improve the performance of the proposed method, instead of searching, one can find a certain formula to be able to specify the right values for e_u or e_d so that the number of white pixels can be changed in the expense of a small modification to the intensity of the pixels. However, as we will show, for an image with the dimensions of 512×512 pixels, by using the proposed procedure, it takes less than a second to embed secret data, which seems to be practical.

4 Implementation Results

Cheng et al. [5] tested various embedding schemes. It is mentioned in [5] that the detection of either DHED or MDHED is more difficult than the schemes that hide data without the availability of the original grayscale image. It is shown that DHED and MDHED schemes produce lower noise like effects as compared to other existing embedding methods.

Based on the results presented in [5], the security of DHED is almost the same as that of MDHED. Therefore, we choose DHED as a successful embedding method and compare its results with those of our proposed SoftSteg method. As an example of our implementations we used the Pills image. Hence, we chose groups of 25 pixels ($L = 25$) and embedded one bit in each group ($N = 1$). This resulted in a stego image with 648 embedded bits. The complexity threshold, T , was chosen to be $2L/5$. This means that the threshold for selecting complex groups is 10. This value was obtained through experimentations. It should be noted that higher values for T leads to lower capacity and higher quality since higher T means embedding in more complex groups.

Figure 3a shows the grayscale version of the pills image [26] while its halftoned version is shown in Figure 3b. To show the halftoning effect we zoomed in a part of the image and increased its contrast. In this image we have plain white and plain black regions along with dark and bright gray regions. Figure 3c shows a SoftSteg stego image with 648 (2.55 percent) embedded bits and Figure 3d shows DHED stego image with the same number of embedded bits.

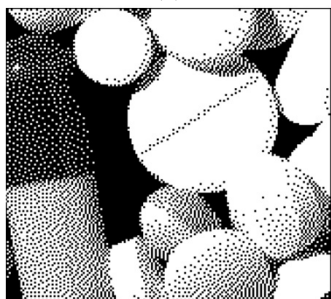
Comparing Figure 3a and Figure 3c we can see that there is no noticeable artifact due to the SoftSteg embedding. In Figure 3d, embedding artifacts are obvious in the black or white regions of the image. Rather than these clear artifacts we can see other artifacts in regions of DHED stego image which were gray in the original grayscale image. As we can see, the visual quality of the proposed method is much better than DHED. To quantify the visual quality, we use the Modified Peak Signal to Noise Ratio (MPSNR) criterion. To find the value of MPSNR [27], we first need to generate the inverse halftone of a stego image. For the inverse halftoning purpose there are several methods mentioned in the literature [28–30]. We apply a Gaussian filter, with the kernel shown in Figure 4, to the halftone image to generate the inverse halftone image. If we find the PSNR of the original grayscale image as compared with its corresponding inverse halftone version then MPSNR is generated.

Figure 5a shows the inverse halftone of the original halftone image. Also, Figure 5b and Figure 5c show

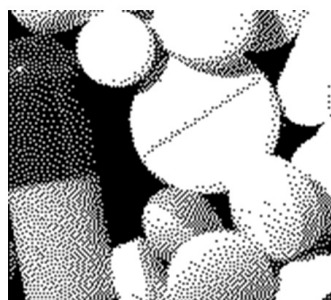




(a)



(b)



(c)



(d)

Figure 3. (a) A 152×167 portion of the original grayscale version of the pills image, (b) Halftoned image using Floyd-Steinberg algorithm, (c) SoftSteg Stego Image with 648 embedded bits, (d) DHED Stego Image with 648 embedded bits.

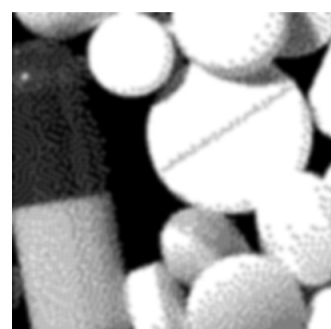
the inverse halftone image of the SoftSteg and DHED stego images respectively.

The artifacts in the totally white or black regions in the inverse halftoned image of DHED stego are clearly present. Furthermore, we can see some artifacts in other regions of the image such as the white spots in

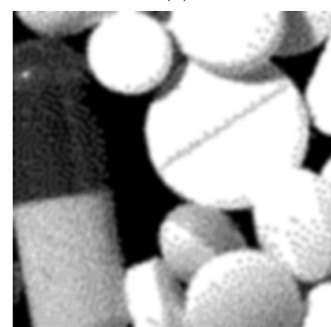
1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

273

Figure 4. The Gaussian filter kernel used for inverse halftoning



(a)



(b)



(c)

Figure 5. (a) Inversed halftone of original halftone image with no embedding, (b) Inversed halftone of SoftSteg stego image, (c) Inverse halftone of DHED stego image.



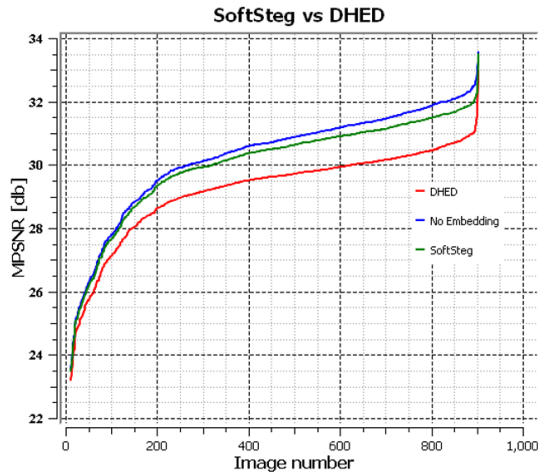


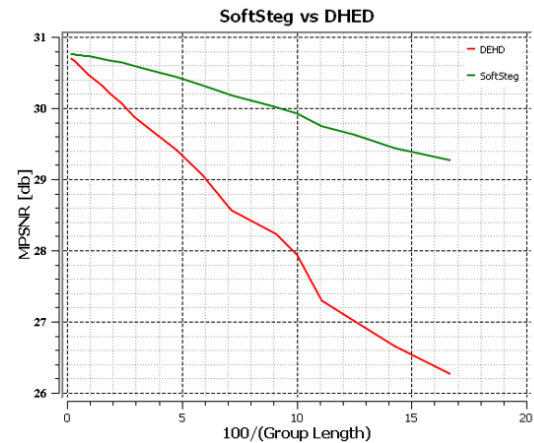
Figure 6. MPSNR of inverse halftoned image of no embedding, DHED, and SoftSteg

the dark side of the capsule. No such artifacts can be seen in the inverse halftone of the SoftSteg stego image. This image is almost identical to the inverse halftone of the cover image.

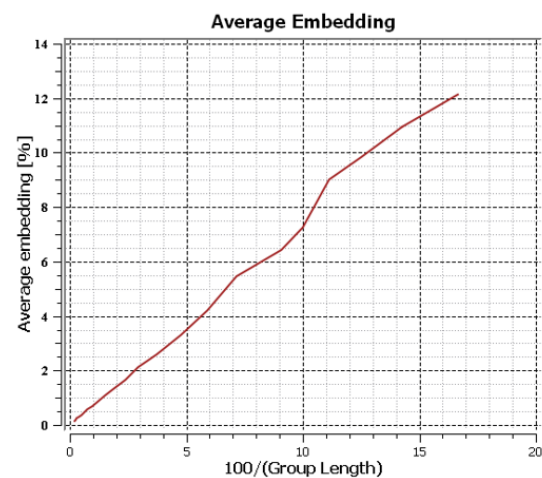
In an experiment, we used 903 natural images and embedded up to 4 percent of the total image size using both DHED and SoftSteg, which means the same number of secret data bits for both methods in each experiment. Then we calculated MPSNR of the inverse halftoned of the stego images as compared to the original gray scale images. Figure 6 shows that the MPSNR curve for the no embedding case has the highest values. The middle line is the curve belonging to the SoftSteg stego images. This curve has values better than those of DHED and very much close to those of images with no embedding.

The average MPSNR of images with no embedding was 30.19 while those of SoftSteg and DHED were 30.11 and 29.67 respectively. In another experiment, same set of images were used for embedding with different group lengths. We embedded 1 bit in each group and the threshold was chosen to be $2L/5$. Figure 7a shows the MPSNR values for the inverse halftoned stego images generated by SoftSteg and DHED.

The horizontal axis shows $100/L$ which is the maximum embedding percent. The maximum group length was chosen as $L = 500$. Hence the smallest value for $100/L$ is 0.2. As the value of L decreases the embedding rate increases. This is shown in Figure 7b. The right axis shows the average embedding percent that is achieved. In Figure 7b, we see that for very low embedding rates the MPSNR values of both SoftSteg and DHED are very close. But as the embedding rate increases and the amount of embedded data grows, the quality of the stego-images of DHED decay very quickly while SoftSteg images are less affected. This is due to the generation of the salt and pepper noise



(a)



(b)

Figure 7. (a): MPSNR of DHED and SoftSteg for different embedding rates, (b): MPSNR of DHED and SoftSteg for different embedding rates.

in the white or black regions of the DHED images. It has been shown that this phenomenon will cause a degradation of the security of DHED.

Another aspect of the proposed algorithm that should be considered, is its time complexity. Using Lena image, with the size of 512 by 512 pixels, we computed the execution time of embedding with various groups lengths. Figure 8 shows the results of the experiment.

Obviously, the increment of the group length increases the searching time for the suitable least change. It will also decrease the number of groups to be searched. As shown in Figure 8 the overall execution time of the proposed method increases with the length of the groups. Anyway, the execution cost is yet reasonable.



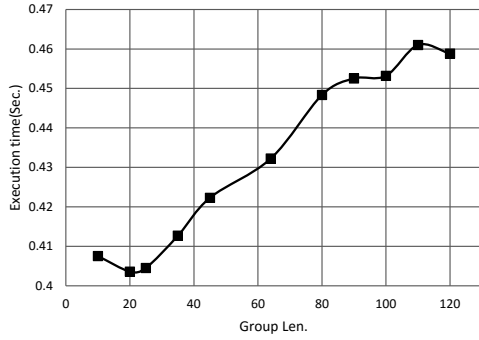


Figure 8. Execution time of the proposed algorithm with various group

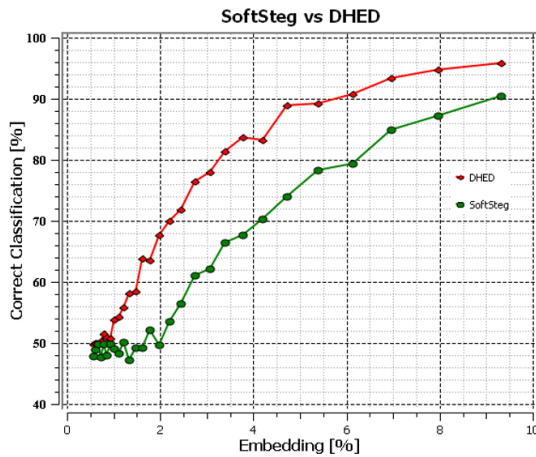


Figure 9. Correct classification percent for DHED and SoftSteg using attack proposed in [5]

5 Security of SoftSteg

To show the security of SoftSteg, we used the steganalysis method proposed in [5]. This steganalysis method is an accurate attack for the halftone images. The security of DHED and MDHED is better than all other methods that are discussed in [5]. Therefore, we will only compare the security of our method with the security of DHED.

In the method of [5], the wavelet statistic features are extracted from the inverse halftone image. These features are fed to a support vector machine (SVM) classifier. Same as in [5], we used the implementation of LIBSVM in our experiments as a powerful two-class classifier. We embedded 7300 random images with different embedding rates using both the SoftSteg and DHED methods. For each rate, we trained a SVM separately for the SoftSteg and DHED and obtained the accuracy of each classification. Using the proposed attack of [5] Figure 9 was produced. These graphs show the correct classification percent for the DHED and SoftSteg methods.

It should be noted that in all comparisons with DHED, same amount of bits are embedded. In another

word, for a 512×512 pixel image, 2% of embedding means $512 \times 512 \times 2/100$ bits which is 5243 bits. Using SoftSteg and for a specified group size, the amount of data bits that can be embedded in an image depends on the context of the image. SoftSteg does not embed in groups that are not complex after halftoning. Data embedding capacities of images that contain large black or white regions are low when the SoftSteg method is applied. Therefore, for a fair comparison between the SoftSteg and DHED, we embedded the same number of bits in an image. Figure 9 indicates that the security of the proposed method is better than DHED because it is less detectable. When the embedding rate is about 2% the correct classification rate for the SoftSteg is about 50% which is equivalent to a random guessing on the part of the classifier. This means that our method is completely secure against the applied steganalysis. For the same embedding rate, the correct classification rate for DHED is 68 percent. This is the security advantage of the SoftSteg over DHED.

6 Conclusion

In this paper we proposed a steganographic method, called the SoftSteg algorithm, for halftone images. We classified the existing steganographic algorithms for the halftone images into two categories. These two classes are those that embed data by altering the halftoning process and those that embed data after an image is halftoned. We introduced our method as the third category that embeds data by altering the grayscale pixels without any change in the halftoning process. We showed that the qualities of our generated stego-images are better than those generated by DHED. It is shown in [5] that the security of DHED is the same as MDHED and is higher than other halftone steganography methods. We also showed that SoftSteg is more secure than DHED while its secure capacity is under 1% comparing to 2% secure capacity of SoftSteg.

Acknowledgements

We would like to thank Dr. Cheng of the Center for Information Security, School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, for providing us with the code that extracts the required features for the steganalytic attack.

References

- [1] Elżbieta Zielińska, Wojciech Mazurczyk, and Krzysztof Szczypiorski. Trends in steganography.



- Communications of the ACM*, 57(3):86–95, 2014.
- [2] Vajihah Sabeti, Shadrokh Samavi, Mojtaba Mahdavi, and Shahram Shirani. Steganalysis and payload estimation of embedding in pixel differences using neural networks. *Pattern Recognition*, 43(1):405–415, 2010.
 - [3] Robert W Floyd. An adaptive algorithm for spatial gray-scale. In *Proc. Soc. Inf. Disp.*, volume 17, pages 75–77, 1976.
 - [4] Thrasyvoulos N Pappas, Jan P Allebach, and David L Neuhoff. Model-based digital halftoning. *Signal Processing Magazine, IEEE*, 20(4):14–27, 2003.
 - [5] Jun Cheng and Alex C Kot. Steganalysis of halftone image using inverse halftoning. *Signal Processing*, 89(6):1000–1010, 2009.
 - [6] Jing-Ming Guo and Yun-Fu Liu. Halftone-image security improving using overall minimal-error searching. *Image Processing, IEEE Transactions on*, 20(10):2800–2812, 2011.
 - [7] Takeshi Ogihara, Shin'ya Koide, and Yukio Kaneda. Data embedding into bilevel images using the error diffusion method. *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, 85(11):36–44, 2002.
 - [8] Min Wu, Edward Tang, and Bo Lin. Data hiding in digital binary image. In *Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on*, volume 1, pages 393–396. IEEE, 2000.
 - [9] Yu-Chee Tseng, Yu-Yuan Chen, and Hsiang-Kuang Pan. A secure data hiding scheme for binary images. *Communications, IEEE Transactions on*, 50(8):1227–1231, 2002.
 - [10] Jing-Ming Guo and Jia-Jin Tsai. Data-hiding in halftone images using adaptive noise-balanced error diffusion. *IEEE MultiMedia*, 2(18):48–59, 2011.
 - [11] Shu-Kei Yip, Oscar C Au, Hoi-Ming Wong, et al. Pi-preserve data hiding for halftone image. In *Intelligent Signal Processing and Communication Systems, 2005. ISPACS 2005. Proceedings of 2005 International Symposium on*, pages 125–128. IEEE, 2005.
 - [12] Ching-Nung Yang, Yao-Yu Yang, Tse-Shih Chen, and Guo-Cin Ye. New steganography scheme in halftone images. In *Intelligent Information Hiding and Multimedia Signal Processing, 2008. IHHMSP'08 International Conference on*, pages 1520–1523. IEEE, 2008.
 - [13] Ming Sun Fu and Oscar C Au. Steganography in halftone images: conjugate error diffusion. *Signal Processing*, 83(10):2171–2178, 2003.
 - [14] Soo-Chang Pei and Jing-Ming Guo. Data hiding in halftone images with noise-balanced error diffusion. *Signal Processing Letters, IEEE*, 10(12):349–351, 2003.
 - [15] Fatemeh Daraee and Saeed Mozaffari. Watermarking in binary document images using fractal codes. *Pattern Recognition Letters*, 35:120–129, 2014.
 - [16] Jeng-Shyang Pan, Hao Luo, and Zhe-Ming Lu. A lossless watermarking scheme for halftone image authentication. *International Journal of Computer Science and Network Security*, 6(2b):147–151, 2006.
 - [17] Ping-Sung Liao, Jeng-Shyang Pan, Yen-Hung Chen, and Bin-Yih Liao. A lossless watermarking technique for halftone images. In *Knowledge-Based Intelligent Information and Engineering Systems*, pages 593–599. Springer, 2005.
 - [18] Ming Sun Fu and Oscar C Au. Data hiding watermarking for halftone images. *Image Processing, IEEE Transactions on*, 11(4):477–484, 2002.
 - [19] Xiaoyi Yu, Yunhong Wang, and Tieniu Tan. Steganalysis of data hiding in binary images. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 4, pages 877–880. IEEE, 2004.
 - [20] Ming Jiang, Xiaolin Wu, Edward K Wong, and Nasir D Memon. Steganalysis of boundary-based steganography using autoregressive model of digital boundaries. In *ICME*, pages 883–886, 2004.
 - [21] Ming Jiang, N Menion, Edward Wong, and Xiulin Wu. Quantitative steganalysis of binary images. In *Image Processing, 2004. ICIP'04. 2004 International Conference on*, volume 1, pages 29–32. IEEE, 2004.
 - [22] Jun Cheng, Alex C Kot, Jun Liu, and Hong Cao. Steganalysis of binary text images. In *ICASSP (4)*, pages 689–692, 2005.
 - [23] Ming Jiang, Edward K Wong, Nasir Memon, and Xiaolin Wu. Steganalysis of halftone images. In *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP'05). IEEE International Conference on*, volume 2, pages ii–793. IEEE, 2005.
 - [24] Siwei Lyu and Hany Farid. Detecting hidden messages using higher-order statistics and support vector machines. In *Information Hiding*, pages 340–354. Springer, 2003.
 - [25] Ming Sun Fu and Oscar C Au. Improved halftone image data hiding with intensity selection. In *Circuits and Systems, 2001. ISCAS 2001. The 2001 IEEE International Symposium on*, volume 5, pages 243–246. IEEE, 2001.
 - [26] Fabien Petitcolas. The information hiding homepage, January 2016. URL http://www.petitcolas.net/watermarking/image_database.
 - [27] Phil Sherry and Andreas Savakis. Improved techniques for watermarking halftone images. In *Acoustics, Speech, and Signal Processing, 2004.*



Proceedings. (ICASSP'04). IEEE International Conference on, volume 5, pages V–1005. IEEE, 2004.

- [28] Li-Ming Chen and Hsueh-Ming Hang. An adaptive inverse halftoning algorithm. *IEEE transactions on image processing*, 6(8):1202–1209, 1997.
- [29] Ping Wah Wong. Inverse halftoning and kernel estimation for error diffusion. *Image Processing, IEEE Transactions on*, 4(4):486–498, 1995.
- [30] Thomas D Kite, Niranjana Damara-Venkata, Brian L Evans, and Alan C Bovik. A fast, high-quality inverse halftoning algorithm for error diffused halftones. *Image Processing, IEEE Transactions on*, 9(9):1583–1592, 2000.



Mojtaba Mahdavi received the B.S. in Computer hardware Engineering From Isfahan University of technology, Iran in 1999. He received the M.S. in Computer architecture from the Isfahan University of Technology in 2002 and his Ph.D. in Electrical Engineering from Isfahan University of Technology, in 2011. He is now an Assistant Professor in the Department of Information Technology, University of Isfahan. His current research interests include Steganography, Steganalysis, Watermarking and Network Covert Channels.



Shadrokh Samavi is a Professor of Computer Engineering at Isfahan University of Technology, Iran and an Adjunct Professor at the ECE department of McMaster University. Professor Samavi completed a B.S. degree in Industrial Technology and received a B.S. degree in Electrical Engineering at California State University, a M.S. degree in Computer Engineering at the University of Memphis and a Ph.D. degree in Electrical Engineering at Mississippi State University, U.S.A.

Dr. Samavi is a Registered Professional Engineer (PE), USA. He is also a member of IEEE and a member of Eta Kappa Nu and Tau Beta Pi honour societies.

Shadrokh Samavi's research interests are in the areas of image processing and hardware implementation and optimization of image processing algorithms. He is also interested in compression and processing of biomedical images, as well as, VLSI design and computer arithmetic.

