A Steganographic Method for Digital Images Robust to RS Steganalysis

André R.S. Marçal and Patricia R. Pereira

Faculdade de Ciências, Universidade do Porto,
DMA, Rua do Campo Alegre, 687, 4169-007 Porto, Portugal

Abstract. Digital images are increasingly being used as steganographic covers for secret communication. The Least Significant Bit (LSB) encoding is one of the most widely used methods for embedding a message in a digital image. However, the direct application of LSB encoding is vulnerable to steganalysis. For example, RS steganalysis is very efficient in detecting the presence of a message in a digital image and to estimate its approximate size. This paper presents a method robust to RS steganalysis, that makes the presence of a message unnoticeable. The method is based on the application of reversible histogram transformation functions to the image, before and after embedding the secret message. The method was tested on 4 greyscale images, with messages of 10%, 30% and 90% of the maximum embedding size. The proposed method proved to be effective in eluding RS steganalysis for all cases tested.

1 Introduction

There has always been interest in safely exchanging secret messages. Throughout times, cryptographic tools have been developed and used in order to make the message exchanged incomprehensible to anyone intersecting the communication. A less known approach is steganography, although also used for thousands of years. Steganography's aim is to make the secret communication undetectable, that is, to hide the presence of the secret message. The recent development of Internet has brought new attention to both cryptography and steganography. The interest in steganography has been enhanced recently by the emergence of commercial espionage and the growing concerns about homeland security due to terrorism. The purpose of steganography is therefore to hide (or embed) a secret message into an artefact, called cover. After embedding the secret information into the cover, it becomes a stego-artefact. Almost anything can be used as cover, as long as it looks common and unsuspicious after the embedding process. With the arrival of the digital era and the generalized usage of the Internet and email for the exchange of files, digital covers such as audio, image and video files have become the most obvious choices. This is party due to their widespread use, but also because this type of media usually includes a random noise component in which the secret message may be easily hidden.

The exchange of ever-growing volumes of data through the Internet, and the widespread access to steganography software prompted the development of

M. Kamel and A. Campilho (Eds.): ICIAR 2005, LNCS 3656, pp. 1192–1199, 2005.

steganalysis tools. The goal of steganalysis is to identify the presence of secret messages embedded in an artefact. An alternative to this passive approach is to perform active or malicious attacks, with the aim of modifying or destroying a secret message that might have been embedded. From the perspective of someone wishing to send a secret message, the best protection is achieved when steganography and cryptography[1] are used together. In this case, even if steganography fails and the presence of the secret message is unveiled, there will still be an additional layer of protection, as the message embedded is ciphered. An overview of the most important topics in criptography, from a steganographer's point of view, is available in [1].

2 Steganography and Steganalysis on Digital Images

Digital images are considered a good choice for a steganography cover because of their insensitivity for the human visual system [2]. The most common approach is to use a substitution system, where the parts of the image considered to be redundant or noisy are replaced by the bits of the secret message [3]. This is done, most of the times, without any change in perceptual content in the cover image [4]. There is no restriction concerning the nature of the message hidden, as long as it can be represented by a stream of bits. In order to recover the message, one only has to know the method and sequence by which the stream of bits was embedded in the image. However, the stego-images produced by these methods may be vulnerable to steganalysis as the embedding process modifies the statistical properties of the cover image. Another problem is that the secret content of the stego-image might be in jeopardy when the image is subjected to manipulations such as lossy compression, denoising or image enhancement, as these operations may modify the content of the image bits, thus making the hidden message recovery impossible.

2.1 LSB Encoding

One of the common data hiding methods is based on manipulating the Least Significant Bit (LSB) planes, by direct replacing the LSBs of the cover image with the message bits [5]. LSB substitution can be performed on all types of images, including raw uncompressed, compressed formats and indexed images. An extended description and discussion of substitution methods, including LSB methods can be found in [3]. The maximum cover size (in bits) is the total number of bytes of the cover image. For 8-bit images, it is the total number of pixels. For example, a text of over 5000 words can be embedded on a 512x512 pixels 8-bit image by the LSB substitution method (maximum message size of 262144 bits = 32768 bytes).

The changes introduced in a 8-bit cover image by the LSB substitution process are very little. The possible swapping of a pixel's LSB results in a change of

[1] Criptography is here used to refer to all the widely used techniques to make information ininteligible to any person who does not hold the key to decipher it.

its Digital Number (DN) by +1 or -1. This will only occur, on average, for every other bit of the embedded message bit stream. Although the use of additional bit planes increases the embedding capacity of an image, the modification in the cover image might become more easily perceptible visually.

An important issue regarding LSB substitution steganography is the selection of the location where the secret message is to be placed in the image. The most common choices are sequential and pseudo-random methods. The sequential method starts on a specific pixel (usually the first one), using all subsequent pixels until the message is fully embedded. The pseudo-random method generate a pseudo-random sequence of values (the indices of bytes of the cover image) that determines the order by which the secret message bits are embedded.

2.2 Visual Steganalysis

One of the most basic forms of steganalysis for digital images is by visual inspection. This can be done directly on the image, or by the observation of a single bit plane, most commonly the LSB. The observation of periodic or other type of suspicions patterns in the image bit plane is used to reveal the presence of a secret message. An example of LSB visual inspection is presented in [6]. Visual steganalysis can be useful in the presence of a small number of suspicious images. However, as the visual inspection of images requires direct human intervention, visual steganalysis is not an effective method for scrutinizing large volumes of image data.

The direct visual steganalysis is based on the faith that the observer's attention will be called out by small discrepancies in uniform regions of the image. When LSB substitution steganography is used, the DN pixel values of the cover image will only have been changed by +1 or -1. For indexed colour images, even this small modification might be easily noticeable, if the variations in colour are not gradual in the colour palette. Colour indexed images are therefore considerably vulnerable to direct visual steganalysis, and should be avoided as cover images for LSB substitution steganography without further processing. On grey scale images, or other type of colour images, a DN change of +1 or -1 will hardly be perceptible, thus the presence of a secret message can only be noticed in very uniform or highly saturated areas.

2.3 Quantitative Steganalysis

Fridrich et al. [7] [8] introduced a powerful, yet complex, steganalytic method that is able to accurately estimate the length of the embedded message on a digital image, for several LSB steganographic methods. The method is based on the fact that the content of each bit plane of an image is correlated with the remaining bit planes. In particular, for an 8-bit image, there is some degree of correlation between the LSB plane and the other seven bit planes. When a message is inserted in the LSB plane, its content is considered to become randomised, and thus the correlation between the LSB plane with the remaining bit planes is reduced or lost.

The RS steganalysis method uses a discrimination function and a flipping operation to identify three types of pixel groups - Regular (R), Singular (S) and Unchanged (U) - depending on how the flipping changes the value of the discrimination function [8]. The size of the group of pixels and the corresponding flipping mask M is initially established. For example M=[010] will correspond to a test performed on groups of 3 pixels at a time, where only the middle one is flipped. In typical images, applying the LSB flipping mask to the pixels in the group will more frequently lead in an increase in the discrimination function, rather than a decrease, and thus the total number of regular groups in an image will be larger than singular groups. The randomisation of the LSB plane forces these differences to zero, as the length of the embedded message increases [8].

When a message with a relative length p is embedded in the cover image ($p = 1$ for full length embedding), the fraction of image pixels with the LSB flipped is, on average, $\frac{p}{2}$. Flipping the LSB of all image pixels will result in an image with a fraction of flipped pixels $1 - \frac{1}{2}$. In the process of steganalysing an image, the actual value of p is unknown. The relative number of R and S groups is counted for the original image, and for the flipped version of that image (with the LSB off all pixels flipped). These will result in four points for the so-called RS diagram, which are used to estimate the value of p. A detailed description of the RS steganalysis can be found in [8]. For greyscale images, the RS steganalysis method can separate cover images from stego images with a 10% embedded message [8].

3 Proposed Method

The standard process consists of embedding a message t into a cover image x. The result is a new image y_1, the stego-image, which should ideally be very similar to the cover image. The process can be represented by equation, where E is the encoding function.

$$y_1 = E(x, t) \tag{1}$$

This process may, however, be vulnerable to steganalytic methods, namely those of statistical steganalysis. The proposed method uses histogram transformation functions in order to defeat steganalysis. The idea is to break the signature that is established between neighbouring pixels in the encoding process, by means of histogram transformation pre and post encoding. Initially, a histogram transformation function f^* is applied to the cover image, x. This function f^* compresses the range of levels used in the original image, to a narrower range. The secret message is embedded in the resulting image $f^*(x)$, producing an image $E(f^*(x), t)$. The final image y_2 is obtained through the application of a histogram transformation function f that expands the range of values back to the initial range, as expressed in Equation (2).

$$y_2 = f(E(f^*(x), t)) \tag{2}$$

A schematic representation of the encoding method is presented in Figure 1. As in the standard method, the final image should be as close as possible to the

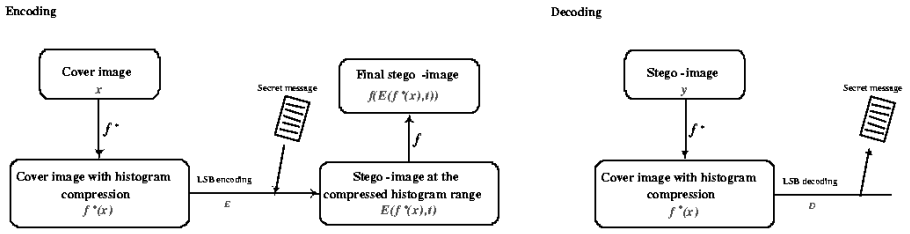

Fig. 1. Scheme of the proposed method

original cover image. In order to extract the message from the image, one has to apply the histogram transformation function f to compress the histogram back to the range of values present at the encoding stage. It is therefore required that the transformation function f^* is the inverse of function f, a property stated in Equation (3).

$$f^*(f(z)) = z \tag{3}$$

Due to the discrete nature of digital images, this property does not stand when the order of function f and f^* is swapped. That is, $f(f^*(z))$ is usually not equal to z. This is not a problem, as long as the changes introduced by the compression and decompression of the histogram are not noticeable. There are other properties that the histogram transformation functions f and f^* should obey to prevent strange artifacts to appear in the image. One requirement is that both functions are monotonous, either increasing (4) or decreasing (5).

$$\forall x_1, x_2;\ x_1 > x_2 \Rightarrow f(x_1) \geq f(x_2) \wedge f^*(x_1) \geq f^*(x_2) \tag{4}$$

$$\forall x_1, x_2;\ x_1 > x_2 \Rightarrow f(x_1) \leq f(x_2) \wedge f^*(x_1) \leq f^*(x_2) \tag{5}$$

In this work, linear histogram transformation functions of the form presented in Equations (6) and (7) were used, where $\lfloor x \rfloor$ is the largest integer below x, and a is a constant. The number of occupied levels of $f(x)$ is the same as in x, but due to the discrete nature of digital images, $f(f^*(x))$ does not occupy the same number of levels as x. Let us consider an example with a cover image of 8-bit unsigned format and $a = 10$. The range of values in the original image x is 0-255, which is reduced to 0-232 by the application of the histogram compression function f^*. The subsequent application of function f (Equation 6) will result in an image y with a range of values 0-255, but with some pairs of levels merged. For example, f^* will compress both levels 219 and 220 to the same value (200), which is then expanded to the level 220.

$$f(x) = x + \left\lfloor \frac{x}{a} \right\rfloor \tag{6}$$

$$f^*(x) = x - \left\lfloor \frac{x}{a+1} \right\rfloor \tag{7}$$

4 Results

Four test images were selected to evaluate the performance of the proposed method. The test images are 512x512 pixels sections of photographs acquired by a digital camera (Peacock, Dinosaurs, Chickens) and a film camera (Falkland). In this later case the film was digitalized in a Kodak Photo Lab. The images were converted to an 8-bit grey scale from their original 24-bit colour format. Figure 2 shows the greyscale version of the test images. Text messages with 10%, 30% and 90% of the maximum embedding size were produced - messages A (3277 characters), B (9830 characters) and C (29491 characters).

Fig. 2. Test images (from left to right): Falkland, Peacock, Dinosaurs, and Chickens

4.1 Message Embedding

The messages were embedded in the cover images using LSB substitution steganography with both sequential and pseudo-random methods. The messages were embedded directly and using the histogram transformation functions (f and f*) with $a = 10$ and $a = 8$ (6,7). A total of 72 stego-images were obtained - 4 cover images, 3 messages, 3 types of functions and 2 location methods. For each test image, 2 additional images were also produced by applying functions f* and f (compression and expansion of the histogram), without embedding any message. A total of 84 images were thus available for testing.

4.2 Visual Tests

The direct visual inspection of the stego-images did not reveal any clues about the presence of the embedded messages, for test images. It is worth pointing out that no direct comparisons were made between the stego-images and the cover images, as in a normal scenario the cover image is not available. In fact, it is good steganography practice to avoid the use of well-known cover images.

A visual inspection of the LSB plane of the stego-images was also performed. This method proved very effective in detecting messages that were directly embedded sequentially. The use of the proposed method reduced the footprint of the sequential embedded message, but a careful observer could still detect the presence of a message. However, the visual inspection of the LSB plane was totally inefficient with pseudo-random LSB substitution steganography.

4.3 RS Steganalysis

The RS steganalysis method was applied to the 72 stego-images, the 4 cover images, and the cover images modified by compression and expansion of the histogram (8 images). Two flipping masks were used: M=[010] and M=[0110]. As no significant differences were obtained with the two flipping masks, the results presented throughout are for M=[010]. The differences in the message size estimation by the RS method between sequential and pseudo-random stego-images were also found to be negligible. Since the stego-images embedded sequentially are vulnerable to visual steganalysis, the results presented are all for pseudo-random LSB encoding.

The message size estimated by RS steganalysis with direct LSB encoding is presented in Table 1. As expected, the RS method proved to be very effective in estimating the message size. The difference between the estimated and the actual length of the embedded message were below 5% for all cases except one - for the test image Falkland and message C (90%) the estimated size was 83%. The results from RS steganalysis with the stego-images produced with the proposed method are considerably different, as it can be seen in Table 2. Two results are presented for each test image and message: using histogram transformation functions (f and f^*) with $a = 10$ and with $a = 8$. When the estimated message length by RS steganalysis is below 5% (the typical range of accuracy of the method), the steganalysis fails to detect the presence of the secret message. This is achieved for all test images and message sizes, although not with every set of histogram transformation functions. For the test image Falkland, the stego-image with message C and functions f / f^* with $a = 10$, the estimated message length is 9.4%. For the test image Peacock, functions f / f^* with $a = 8$ produce high negative values for the estimated message length, even without any

Table 1. Estimated message length for standard (direct) encoding

MESSAGE	FALKLAND	PEACOCK	DINOSAURS	CHICKENS
None	0.1%	3.0%	1.3%	-1.4%
A (10%)	9.7%	12.9%	12.1%	9.5%
B (30%)	29.1%	31.8%	31.8%	29.6%
C (90%)	83.0%	87.4%	85.4%	89.3%

Table 2. Estimated message length for the proposed method of encoding

MESSAGE	FALKLAND		PEACOCK		DINOSAURS		CHICKENS	
	$a = 10$	$a = 8$	$a = 10$	$a = 8$	$a = 10$	$a = 8$	$a = 10$	$a = 8$
None	4.8%	-1.2%	1.0%	-16.7%	-1.6%	3.3%	-0.3%	-1.3%
A (10%)	5.0%	0.7%	0.9%	-18.8%	-0.7%	4.3%	-1.6%	-1.1%
B (30%)	4.3%	-0.5%	0.9%	-21.7%	-0.7%	3.7%	-0.7%	-0.3%
C (90%)	9.4%	4.9%	0.5%	-29.5%	-3.6%	2.0%	2.7%	2.3%

message. The results for the other two test images are all below the 5% mark for both sets of functions.

5 Conclusions

The proposed method, based on the application of reversible histogram transformation functions to the cover image, proved effective in defeating RS steganalysis. The RS steganalysis method was successful in estimating the message length embedded in standard stego-images, but failed to detect the presence of the embedded messages, when using the proposed method. The two types of functions used were effective on two test images, with messages of 10%, 30% and 90% of the maximum embedding length. On the other two test images, only one of the functions performed well for all 3 messages. The choice of the adequate set of histogram transformation functions is dependent of the cover image. The implementation of the proposed method as a robust steganography software would, therefore, require the automatic testing of various sets of histogram transformation functions, for a given cover image, selecting the set that provides the best results for that image. An indication of the histogram transformation functions used would then have to be inserted in a pre-defined location in the cover image. For example, the linear histogram transformation functions used here would be characterised by a single byte (the value of the parameter a).

References

1. Eric Cole, Hiding in Plain Sight: Steganography and the Art of Covert Communication. Wiley Publishing, Inc., (2003)
2. Lou, D.C., Liu, J.L.: Steganographic method for secure communications. Computers and Security, **21** (2002) 449–460
3. Katzenbeisser, S., Petitcolas, F.A.P.: Information Hiding Techniques for Steganography and Digital Watermarking. Artech House (2000)
4. Chandramouli, R., Memon, N.: Analysis of LSB based image steganography techniques. Proceedings of the IEEE International Conference on Image Processing. **3** (2001) 1019–1022
5. Chan, C.K., Cheng, L.M.: Hiding data in images by simple LSB substitution. Pattern Recognition, **37** (2004) 469–474
6. Westfeld, A., Pfitzmann, A.: Attacks on steganographic systems. Lecture Notes in Computer Science, Springer-Verlag **1768** (2000) 61–75
7. Fridrich, J., Goljan, M., Rui Du: Detecting LSB steganography in color, and gray-scale images. IEEE Multimedia **8** (2001) 22–28
8. Fridrich, J., Goljan, M., Hogea, D., Soukal, D.: Quantitative steganalysis of digital images: estimating the secret message length. Multimedia Systems **9** (2003) 288–302