

SLSB: Improving the Steganographic Algorithm LSB

Juan José Roque, Jesús María Minguet
Universidad Nacional de Educación a Distancia (Spain)
juanjose.roque@extremadura.es; jminguet@issi.uned.es

Abstract. This paper presents a novel steganographic algorithm based on the spatial domain: Selected Least Significant Bits (SLSB). It works with the least significant bits of one of the pixel color components in the image and changes them according to the message's bits to hide. The rest of bits in the pixel color component selected are also changed in order to get the nearest color to the original one in the scale of colors. This new method has been compared with others that work in the spatial domain and the great difference is the fact that the LSBs bits of every pixel color component are not used to embed the message, just those from pixel color component selected.

Keywords: Security, Steganography, Least Significant Bit.

1 Introduction

The steganography can be considered as a branch of cryptography that tries to hide messages within others, avoiding the perception that there is some kind of message. To apply steganographic techniques cover files of any kind can be used, although archives of image, sound or video files are the most used today. Similarly, information to hide can be anything: text, image, video, sound, etc.

There are two trends at the time to implement steganographic algorithms: the methods that work in the spatial domain (altering the desired characteristics on the file itself) and the methods that work in the transform domain (performing a series of changes to the cover image before hiding information. To select the best areas the Discrete Cosine Transform DCT, Wavelet Transform, etc. are used).

While the algorithms that work in the transform domain are more robust, that is, more resistant to attacks, the algorithms that work in the spatial domain are simpler and faster.

The best known steganographic method that works in the spatial domain is the LSB [1] (Least Significant Bit), which replaces the least significant bits of pixels selected to hide the information. This method has several implementation versions that improve the algorithm in certain aspects [2][3][4][5][6][7].

This paper proposes a new method, SLSB (Selected Least Significant Bit), that improves the performance of the method LSB hiding information in only one of the three colors at each pixel of the cover image. To select the color it uses a Sample Pairs analysis, given that this analysis is more effective to detect hidden information. Finally, applies a LSB Match [8] method so that the final color is as close as possible to the original one in the scale of colors.

The paper is organized as follows. Section 2 gives a brief classification of the steganographic methods that works in spatial domain. Section 3 describes the

proposed method. Section 4 is on the experimental results, followed by conclusions at Section 5.

2 Methods in Spatial Domain

A basic classification of steganographic algorithms operating in the spatial domain as the method for selecting the pixels distinguishes three main types: non-filtering algorithms, randomized algorithms and filtering algorithms.

2.1 Non-filtering Algorithm

This is the simplest steganographic method based in the use of LSB, and therefore the most vulnerable. The embedding process consists of the sequential substitution of each least significant bit of the image pixel for each bit of the message. For its simplicity, this method can camouflage a great volume of information [9].

This technique is quite simple. It is necessary only a sequential LSB reading, starting from the first image pixel, to extract the secret message. This method also generates an unbalanced distribution of the changed pixels, because the message is embedded at the first pixels of the image, leaving unchanged the remaining pixels.

2.2 Randomized Algorithm

This technique was born as a solution for the problems of the previous method. Each one, the sender and the receiver of the image has a password denominated stego-key that is employed as the seed for a pseudo-random number generator. This creates a sequence which is used as the index to have access to the image pixel. The message bit is embedded in the pixel of the cover image as the index given by the pseudo-random number generator [9].

The two main features of the pseudo-random permutation methods are the use of password to have access to the message, and the well-spread message bits over the image.

2.3 Filtering Algorithm

This algorithm filters the cover image by using a default filter and hides information in those areas that get a better rate. The filter is applied to the most significant bits of every pixel, leaving the less significant to hide information. The filter ensures the choice of areas of the image in the least impact with the inclusion of information, which affects a greater difficulty of detecting the presence of hidden messages [10]. The retrieval of information is ensured because the bits used for filtering are not changed, implying that the reapply the filter will select the same bits in the process of concealment. It is the most efficient method to hide information.

The algorithm SLSB belongs to this type.

3 Description of the Algorithm SLSB

Figure 1 shows the structure of the algorithm SLSB:

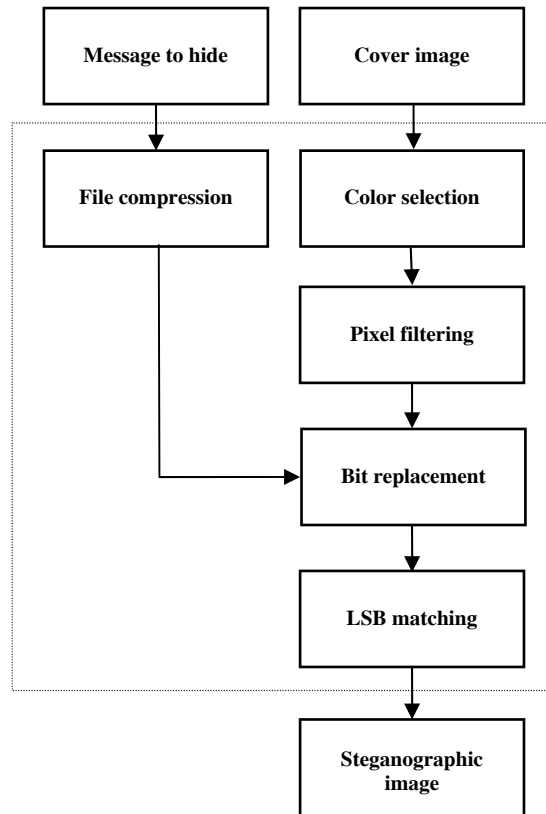


Fig. 1. Structure of the algorithm SLSB.

3.1 Hiding Information in Only One Color

Most of the algorithms that work in the spatial domain using a LSB method (or any of its derivatives) as the algorithm for information hiding, that is, hide one bit of information in the least significant bit of each color of a pixel.

But these methods can't stand a type of statistical analysis (such as RS [11] or Sample Pairs [12]), even if partly camouflaged in the amount of information hidden. The problem stems from the fact that modifying the three colors of a pixel produces a major distortion in the resulting color. This distortion is not visible to the human eye, but detectable by statistical analysis.

For example, if a pixel of the cover image with the RGB (Red-Green-Blue code) color A8A8A8 # is used, binary 10101000-10101000-10101000, and 1 bit with value 1 is set on each LSB bit of each color component, to hide the message 111, the result would be 10101001-10101001-10101001:

Table 1. Results obtained hiding the message 111 in the pixel 10101000-10101000-10101000 with the LSB method.

	Hexadecimal	Decimal	Red	Green	Blue
Original pixel	A8A8A8	11053224	168	168	168
Modified pixel	A9A9A9	11119017	169	169	169

In theory the three least significant bits of the pixel have changed, introducing a small distortion, but the difference between the old and new color represents a leap of 65793 colors in the scale of colors.

One method that would introduce more efficiency and less distortion would store the 3 bits of information to hide in the same color. Using the same example, the 3 bits of information will be introduced in the 3 LSB bits of green color (10101000-10101**111**-10101000):

Table 2. Results obtained hiding the message 111 in the pixel 10101000-10101000-10101000 with the SLSB method.

	Hexadecimal	Decimal	Red	Green	Blue
Original pixel	A8A8A8	11053224	168	168	168
Modified pixel	A8AFA8	11055016	168	175	168

In this case the leap in the scale of colors is 1792 colors (in the case of changing the color green, if modify the blue color difference would be only 7 colors), that being the extreme case because it has been replaced last 3 bits with 0 value for 3 bits with a 1 value, that is, in most cases the distortion will be much lower.

In order to choose the color for the concealment, the SLSB algorithm performs a preliminary Sample Pairs analysis and select the color with higher ratio because it represents more diversity, leading to less noticeable changes. The choice of Sample Pairs analysis over other steganalitics methods is due to the results provided by the work of Ker [13], where this analysis shows that it is offering better results in terms of detecting hidden information. Thus, the chosen color will be the one that provides greater distortion and, therefore, the result of the withholding of information will be less detectable.

3.2 LSB Match Adaptation

Following the work of Van Dijk [14] and Goljan [15], the method LSB Match (designed to work with a single LSB bit) has been adapted to allow an LSB Match with any number of LSB bits.

This method calculates the distance between the original color and the steganographic color. Should the distance is greater than a certain threshold (determined by the number of bits to hide) the color is decremented to get a final color closest to the original, implying a further reduction in the distortion caused by the hidden information.

For example, using a cover byte 11001000 to hide 3 bit of information (111), with a simple LSB results in 11001111, which has a difference of 7 values with respect to the original.

Applying the method proposed here to the above example (in this case, decreasing the 4th least significant bit, which have been used 3 bits LSB to hide information) results in 11000111, with a distance of 1 from the original byte but with the same hidden information.

4 Results

To be able to compare the performance of this improvement on the LSB method, the image on Fig. 1 will be used as cover with BMP (Bit Mapped Picture) format and 512x512 pixels in size (24 bits/pixel).



Fig. 2. Cover image.

4.1 Histogram Analysis

The purpose of the histogram analysis is to detect significant changes in frequency of appearance of the colors by comparing the cover image with the steganographic image.

To better align this analysis it has been carried out a detailed examination of the 4 components of any image: brightness and red, green and blue colors.

Histograms in Fig. 3 shows a frequency histogram of the image on Fig. 2 for the four components mentioned above, and his comparison with the results of the image on Fig. 2 with a hidden message of 141.744 bits, using a hiding method of 1 bit/pixel, producing a hiding rate of 54%.

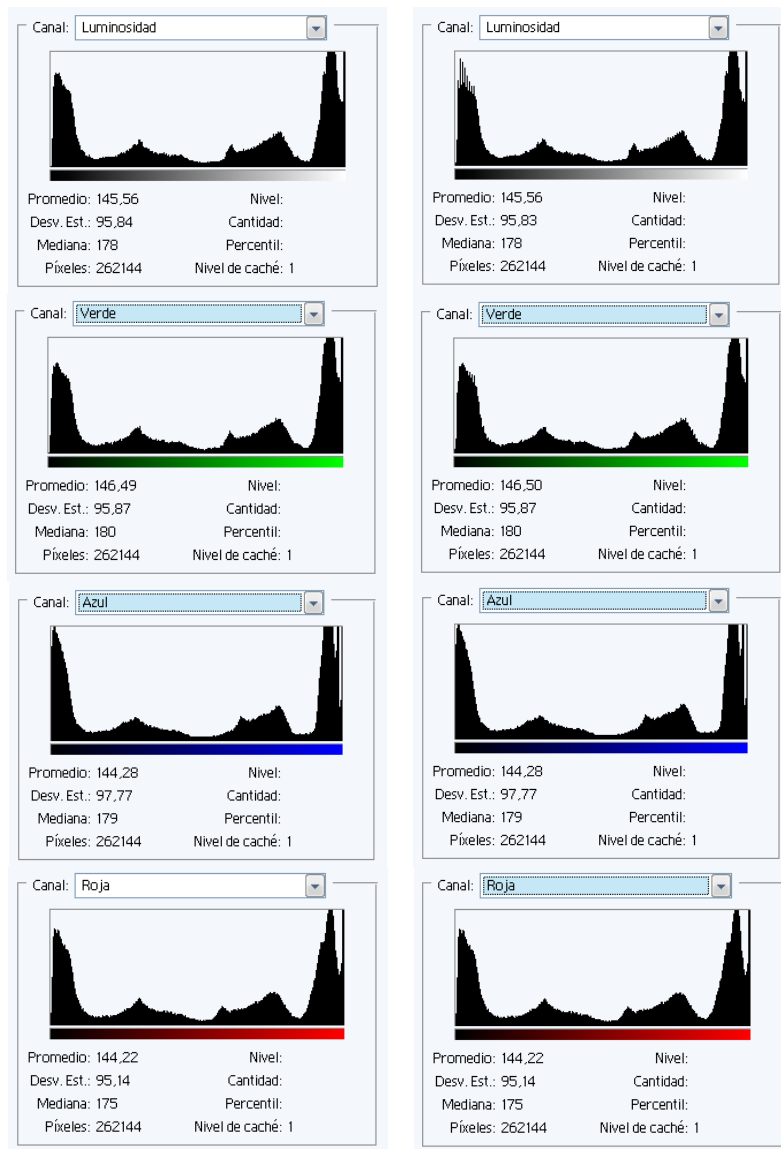


Fig. 3. Histograms of brightness, green, blue and red colors in the image on Fig. 2 (left) and the comparison with the results of the image on Fig. 2 with a hidden message of 141.744 bits, using a hiding method of 1 bit/pixel, producing a hiding rate of 54% (right).

There are only changes in the histograms of brightness and green color (the one chosen by the algorithm as the optimal color of concealment).

Despite having a hiding rate of 54%, the changes are negligible (0.01 in the standard deviation of brightness and an average of 0.01 in the green color).

According to the results, it can be said that the new proposed algorithm is immune

to attacks based on a comparison of histograms of the original image and the steganographic image.

4.2 Another Steganographic Tools Comparison

To conclude the analysis of the results of the new proposed algorithm its performance is compared with that from the best known and more used today steganographic tools.

This comparison focuses on two aspects: the results of the RS and Sample Pairs analysis of steganographic images and the analysis of the results of the metrics of distortion.

Table 3 shows a comparison of the results for the steganographic images obtained with the various tools in front of the RS analysis and the Sample Pairs analysis. The results of these analyses are an estimate of the percentage of hidden information. A lower ratio means a higher quality of the hiding method.

Table 3. Results obtained using a cover image of 786.486 bytes (Fig. 2) and a hidden message of 31.071 bytes (TXT file).

Tool	RS analysis	Sample Pairs analysis
Hermetic Stego [20]	75,46911	73,39835
Invisible Secrets [23]	70,06539	69,32617
Hide4PGP [21]	30,60135	30,19531
Contraband [16]	14,78796	11,83324
wbStego [27]	14,33760	13,42652
White Noise Storm [28]	11,91106	10,12546
Digital Invisible Ink Toolkit [18]	9,61806	7,84342
JPHS [24]	2,62679	2,68511
S-tools [26]	2,30629	2,10435
EikonaMark [19]	1,86631	1,31909
Data Privacy Tools [17]	1,43103	0,96443
Hide In Picture [22]	1,03530	1,06373
SLSB algorithm (1 bit/pixel)	0,89172	0,61744
SLSB algorithm (2 bits/pixel)	0,80084	0,60556
Original image	0,67766	0,51907
SLSB algorithm (3 bits/pixel)	0,64431	0,47867
Steghide [25]	0,63543	0,37671

The results show that the new algorithm, in its 3 versions, offers among the best ever results (even reach a ratio below the original image, thereby preventing distinction between the original and steganographic image).

Table 4 shows a comparison of the results of the metrics of distortion [9] (Average Absolute Difference, Mean Squared Error, Lp-Norm, Laplacian Mean Squared Error, Signal to Noise Ratio, Peak Signal to Noise Ratio, Normalised Cross-Correlation and Correlation Quality) applied to steganographic images obtained by different tools. Obviously, a lower distortion represents a better steganographic method because it is closer to the values of the original image.

Table 4. Results obtained using a cover image of 786.486 bytes (Fig. 2) and a hidden message of 31.071 bytes (TXT file).

Tool	AAD	MSE	LP	LMSE	SNR	PSNR	NC C	CQ
Original image	0,0	0,000	0,0	0,000	0,0	0,0	1,000	1,26643
SLSB algorithm (1 bit/pixel)	5,3	0,020	36,9	1,113	6,7	2,0	0,999	1,26643
SLSB algorithm (2 bits/pixel)	6,1	0,043	53,2	2,279	3,2	9,9	1,000	1,26643
SLSB algorithm (3 bits/pixel)	9,1	0,137	94,9	7,288	1,0	3,1	1,000	1,26643
Contraband	10368,4	0,872	26038,6	0,002	136124,5	415146,6	0,999	1,26597
Data Privacy Tools	25155,1	13,760	110709,9	0,045	7530,1	22965,0	1,001	1,26798
Digital Invisible Ink Toolkit	10377,1	0,947	26053,1	0,002	135973,3	414685,4	1,000	1,26662
EikonaMark	150528,2	90,996	232052,0	0,279	1713,9	5227,1	0,997	1,26307
Hermetic Stego	42354,0	4,500	52662,9	0,009	33278,5	101491,3	1,000	1,26643
Hide4PGP	10523,7	0,477	26211,1	0,002	134339,1	409701,5	1,000	1,26643
Hide In Picture	10493,4	0,665	26194,6	0,002	134508,4	410218,1	1,000	1,26747
Invisible Secrets	32934,0	3,007	46420,3	0,007	42830,9	130623,8	1,000	1,26649
JPHS	54609,9	9,141	89871,2	0,011	11427,0	34849,6	1,000	1,26643
Steghide	1465,3	0,320	16270,8	0,001	348621,5	1063210,6	0,999	1,26643
S-tools	552,4	0,025	6005,0	1,202	2559393,7	7805527,2	1,000	1,26643
wbStego	10408,3	0,947	26092,5	0,002	135563,4	413435,6	1,000	1,26668
White Noise Storm	13828,4	1,101	30071,6	0,003	102060,9	311260,8	0,999	1,26640

This table can verify that the new algorithm (in any of its three versions) offers the best results in the metrics AAD, MSE, LP, SNR, PSNR, NCC and CQ, and provide the same result as the original image in the last two columns.

5. Conclusions

This paper proposes a new method, SLSB (Selected Least Significant Bit), that improves the performance of the LSB method hiding information in only one of the three colors at each pixel of the cover image. For the selection of color it uses a Sample Pairs analysis, given that this analysis is more effective to detect hidden information. Finally, applies a LSB Match [8] method so that the final color is as close as possible to the original one.

A summary of its features could be:

- It is based on the LSB method, but can hide the same information much more effectively using bits of just one color.
- Perform a Sample Pairs analysis prior to steganography, which allows you to select the best color of the three possible to hide information.
- Use a pixel selection filter to obtain the best areas to hide information.
- Implement the LSB Match method to reduce the difference between the original pixel and the steganographic pixel.
- It is immune to visual attacks. Changes are undetectable with the naked eye, and a filter of LSB bits doesn't present areas of random information that could indicate the presence of hidden information.
- It is immune to attacks by comparing histograms, as the frequency of appearance of colors in the steganographic image is very similar to that of the cover image.
- It is immune to statistical attacks, as two colors for each pixel are equal to those of the original image, and the final ratio of analysis is very close to the original image, which doesn't raise suspicion it contains hidden information. Even in some cases get better rates than those of the original image, creating confusion over which of two images would be the original.
- It yields well above that of most steganographic tools used today, both in RS and Sample Pairs analysis and in metric of distortion.

Future works will aim to achieve better performance and be undetectable by the most famous steganographic analysis, for example, changing bits undisturbed by the concealment of the message.

References

- 1 Kurak, C. and McHugh, J.: A Cautionary Note on Image Downgrading. Proc. IEEE 8th Annual Computer Security Applications Conference. San Antonio, USA, Nov./Dec. 1992, pp. 153-155.
- 2 Moskowicz, I., Longdon G. and Chang, L.: A New Paradigm Hidden in Steganography. Proc. 2000 Workshop on new security paradigms, Ballycotton, Country Cork, Ireland, 2000. ACM Press, New York, pp. 41-50.
- 3 Sharp, T.: An implementation of key-based digital signal steganography. Proc. 4th International Workshop on Information Hiding, Pittsburgh, USA, April 25, 2001. Springer LNCS, vol. 2137, pp. 13-26.

- 4 Kawaguchi, E. and Eason, R.: Principle and applications of BPCS-Steganography. Proc. Multimedia Systems and Applications Conference, Boston, MA, USA, November 2, 1998. SPIE series, vol. 3528, pp. 464-473.
- 5 Bender, W., Gruhl, D., Morimoto, N. and Lu, A.: Techniques for data hiding. IBM Systems Journal, vol. 35, nos. 3&4, 1996.
- 6 Moskowitz, I., Johnson, N. and Jacobs, M.: A detection study of an NRL steganographic method. NRL Memorandum Report NRL/MR/5540{02-8635, Naval Research Laboratory, Code 5540, Washington, 2002.
- 7 Noto, M.: MP3Stego: Hiding text in MP3 files. Sans Institute, 2003.
- 8 Sharp, T.: An implementation of key-based digital signal steganography. Proc. 4th International Workshop on Information Hiding. Springer LNCS, vol. 2137, pp.13-26, 2001.
- 9 Katzenbeisser, S. and Petitcolas, F.: Information hiding techniques for steganography and digital watermarking. Artech House Books, 1999.
- 10 Hempstalk, K.: Hiding behind corners: using edges in images for better steganography. Computing Womens Congress Conference, Hamilton, New Zealand, 2006.
- 11 Fridrich, J., Goljan, M. and Du, R.: Reliable detection of LSB steganography in color and grayscale images. Proc. ACM Workshop on Multimedia and Security, Ottawa, ON, Canada, Oct. 5, 2001, pp. 27-30.
- 12 Dumitrescu, S., Wu, X. and Wang, Z.: Detection of LSB steganography via sample pairs analysis. 5th International Workshop on Information Hiding. Noordwijkerhout, Pays-Bas, 7/10/2002. Springer LNCS, vol. 2578, pp. 355-372, 2003.
- 13 Ker, A.: Improved detection of LSB steganography in grayscale images. Proc. 6th Information Hiding Workshop. Springer LNCS, vol. 3200, pp. 97-115, 2004.
- 14 Van Dijk, M. and Willems, F.: Embedding information in grayscale images. Proc. 22nd Symposium on Information and Communication Theory in the Benelux, pp. 147-154, Enschede, The Netherlands, May 15-16, 2001.
- 15 Goljan, M. and Holotyak, T.: New blind steganalysis and its implications. Proc. SPIE Security, Steganography, and Watermarking of Multimedia Contents VIII, vol. 6072, pp. 1-13, 2006.
- 16 Thyssen, J. & Zimmerman, H., Contraband 9g, 1999-01-01 (consultation date 2009-07-07). Available in <http://www.jthz.com/puter/>.
- 17 Bernard Electronics, Data Privacy Tools 3.5, last update 2008-10-03 (consultation date 2009-07-07). Available in http://www.xs4all.nl/~bernard/home_e.html
- 18 Hempstalk, K., Digital Invisible Ink Toolkit 1.5, last update 2006-06-09 (consultation date 2009-07-07). Available in <http://diit.sourceforge.net>
- 19 Alpha Tec Ltd, EikonaMark 4.7, last update 2005-01-01 (consultation date 2009-07-07). Available in <http://www.alphatecltd.com/watermarking/eikonamark/eikonamark.html>

- 20 Hermetic Systems, Hermetic Stego 7.41t, last update 2008-08-09 (consultation date 2009-07-07). Available in <http://www.hermetic.ch/hst/hst.htm>
- 21 Repp, H., Hide4PGP 2.0, last update 200-02-01 (consultation date 2009-07-07). Available in <http://www.heinz-repp.onlinehome.de/Hide4PGP.htm>
- 22 Figueiredo, D., Hide In Picture 2.1, last update 2002-10-01 (consultation date 2009-07-07). Available in <http://sourceforge.net/projects/hide-in-picture>
- 23 Neobyte Solutions, Invisible Secrets 4.6.3, last update 2007-04-01 (consultation date 2009-07-07). Available in <http://www.invisiblesecrets.com>
- 24 Latham, A., JPHS 0.5, last update 1999-08-01 (consultation date 2009-07-07). Available in <http://linux01.gwdg.de/~alatham/stego.html>
- 25 Hetzl, S., StegHide 0.5.1, last update 2003-10-15 (consultation date 2009-07-07). Available in <http://steghide.sourceforge.net>
- 26 Brown, A., S-tools 4, last update 200-11-18 (consultation date 2009-07-07). Available in <http://www.spychecker.com/program/stools.html>
- 27 Bailer, W., wbStego 4.3, last update 2004-03-01 (consultation date 2009-07-07). Available in <http://www.8ung.at/wbailer/wbstego/>
- 28 Arachelian, R., White Noise Storm 2.10, last update 1995-02-01 (consultation date 2009-07-07). Available in <http://www.nic.funet.fi/pub/crypt/steganography/>