A vulgarization article about steganography, illustrated by a simple demonstration of the vulnerability of the LSB substitution method

J. Rabault
correspondence to the author: jean.rblt@gmail.com

18th August 2017

While the aim of encryption is to make a message impossible to decipher to an attacker, the aim of steganography is to hide that exchange of information - other than the innocuous cover - is taking place. A common support for steganography is digital images, and many methods have been proposed for performing steganography on such supports. However, some of the simplest steganography methods are well known to be vulnerable to attacks that can reveal that data is hidden, and even give an estimate of how big the message is. In this paper, we use a very simple approach to illustrate the well-known vulnerability of the Least Significant Bit substitution (LSB-substitution) method, and explain why the LSB-matching approach, while not immune to attacks, is a safer alternative.

Disclaimer 1: this paper is more about vulgarization and dissemination of simple ideas about steganography, than about reporting new research. As the primary goal is vulgarization, I feel free to relax some constraints I would apply on an article or article draft and therefore to be more informal. Also parts of the discussion may be simplistic or incomplete, and some facts may be presented without due references to the literature.

Disclaimer 2: steganography is both a technology of potential importance for enforcement of privacy and an interesting theoretical problem with no easy answer, therefore making it an interesting intellectual challenge. However, nothing that is presented here comes with any guarantee. In addition any technology can be used in an illegitimate way. I strongly advice you to not use steganography to try to break the law, and I decline all responsibility if any idea presented here should be used to try to break the law.

1 Introduction

1.1 Cryptography and steganography

Cryptography is the cornerstone of much of the internet. Cryptography allows to encrypt a message in such a way that, if performed correctly, only 'legitimate' actors can decipher it in a reasonable amount of time. Hiding the meaning of a message has been a critical capability for war, intelligence, and other (sometimes illegal) activities of various types since, at least, antiquity. For example, a simple cypher algorithm is named after Julius C. Caesar. However, while cryptography hides the content of the message, cryptography does not hide that exchange of information is taking place. This proves critical in many real-world applications, when the simple knowledge that encrypted communication is taking place between actors is enough to map networks, or gain information about which kind of services are exchanged or which kinds of protocols are used, therefore increasing dramatically the attack surface - not to talk about being a possible motivation for exerting constraints, legal or physical, in view of obtaining the decrypted message.

As a consequence, hiding that communication is taking place is also a capability of strategic importance. This is the aim of steganography, that hides messages into seemingly innocuous appearances. There also, the strategic importance of steganography has motivated the development of real-world implementations since at least antiquity (the classical example being, a king in ancient Greece wrote messages on the shaved head of slaves, and waited for their hair to cover the message before sending them). Of course, the best approach is to use both cryptography and steganography together so that, even if the steganography cover gets discovered, the message cannot be deciphered (and it is also probably quite difficult to prove, on a legal level, that a message is actually hidden without deciphering it as most steganography detection methods are only statistical, and can have false positive and negative). In addition, well-behaved cryptography turns any structured message into a seemingly random sequence, therefore breaking some of the structure that could be the target of attacks against steganography. The specific task of identifying whether or not a message is hidden in apparently innocuous data is called steganalysis.

Cryptography and steganography are made even more important in the age of digital information, due to the increasing role of communication networks in all aspects of life. In addition, modern computers, algorithms and hardware make it easier than ever to implement such techniques. However, both cryptography and steganography remain subjects of open research, and probably admit no definite solutions but rather are the theater of a forever-raging war between the sword and the shield.

A good illustration of this last point is the fact that encryption methods (and related, such as hash functions) get in, and out, of fashion as vulnerabilities are found. Even mathematically sound methods need to take into account the exponentially increasing power of modern computers, and new recommendations are issued regarding the length of the encryption keys to use. Due to the importance of cryptography, those challenges regarding cryptography are well known also for the educated non specialist and information on this matter, as well as guidelines, are easy to find.

By contrast, this issue is a bit less highlighted in the case of steganography as its use is more confidential, and the non-specialist may be less aware of steganography methods becoming vulnerable to attacks. Moreover, a lot of the literature about steganalysis is hard to read and one could argue that little vulgarization is available. In this paper, we want give one (more) simple illustration of the well-known vulnerability of the LSB-substitution method and explain quickly why, while not immune to attacks, LSB-matching is a safer method.

The organization of the paper is the following. The next paragraphs of this section are a short literature review of some key aspects of modern steganography. The next section is a simple illustration of the vulnerability of LSB-substitution, together with an explanation of why such effective detection methods can be applied to LSB-substitution. Finally, a few words of conclusions are issued. Globally, the aim of this paper is to vulgarize ideas around steganography and steganalysis, and make it clear that steganography must be done properly to work.

1.2 A short literature review

In the following paragraphs of this section, we present a short review of the literature about steganography and steganalysis. Steganography methods can be applied to virtually any type of noisy support on which part of the data can be altered without letting traces, such as digital images, videos, audio records, text data, spam messages [1], timing of requests between servers, etc. However in the following we limit ourselves to the case of digital images.

Steganography on digital media has been known for several decades, and both steganography and steganalysis are heavily discussed in the literature. For a good introduction, the reader is referred to [2, 3].

The development of steganography has followed the development of digital media technologies. In the early days of the internet, GIFs were used to perform steganography due to their abundance. However, the way information is stored on GIFs makes it very easy to perform steganalysis, and the use of either .png or .jpg for embedding information is seen as a better alternative [2, 4].

Several simple methods can be used to perform steganography on lossless compression bitmap formats, such as png or tiff. The simplest, know as LSB-substitution, consists in simply replacing the least significant bit of R, G, and B channels by the data to hide. Which pixels and channels to use as slots for storing information is decided from an encryption key. The encryption key can for example be used as the seed of a pseudo-random number generator which generates a pseudo random (but deterministic) list of slots, that can be used for writing and reading the message. This method, while simple to implement, suffers from several flaws that make it very easy to detect even in real-world applications. For example the so-called RS-steganalysis is a well-known method that is able to accurately reveal if, and how much, information is embedded using LSB-substitution [2, 5, 6, 7]. As a consequence, LSB-matching was developed in an attempt to make LSB steganography more robust. In the case of LSB-matching, the value of the pixel channel to use for embedding is modified by ± 1 (which sign being used is chosen randomly, both are given a probability 0.5) if it does not correspond to the bit to hide. This seemingly minor change in the method makes LSB-matching significantly more difficult to detect than LSB-substitution, to the point where, while steganalysis methods attacking LSB-matching have been presented in the literature, their performance on real-world data is still moderate and the design of attack methods is still an active research topic [8, 9, 10, 11, 12, 13]. In addition, a number of enhancement to LSB-matching have been proposed in the literature, that make its detection even more challenging [14, 15, 16].

However, images encoded using lossless compression algorithms cannot be regarded as the dominant norm on the internet (and in much of the modern digital world). Indeed, most often one does not need the full level of details of a multi-megapixel image and therefore one is willing to use lossy compression to reduce the size of images. The very popular jpg (or jpeg) format, is such a lossy compression format that takes advantage of the characteristics of the

human perception system to design an algorithm that can significantly compress images, with moderate visual impression of quality loss (at least, as long as a reasonable compression factor is used). However, performing steganography on such supports requires extra caution as the compression algorithm introduces new structures in the image, and therefore one must be careful not to disturb them (since the compression and decompressino algorithms are well known, disturbing those structures could be detected by the steganalyst). In particular, a technique as simple as LSB steganography cannot be applied on jpeg images (or images that have previously been stored as jpeg) as this would create 'nearly jpeg' images, i.e. images where the existence of jpeg artifacts is visible while the image is not any longer the result of jpeg decompression. As a consequence, one needs to alter the compressed data used for generating the decompressed image rather than the decompressed bitmap in order to perform steganography on jpeg. This proves difficult, and while a number of methods were presented, a modern and well-known such method being F5 [17], vulnerabilities to steganalysis have been described [18, 19]. As a consequence, it is probably wise to avoid as a support jpg, jpeg, or other lossy compression images, or any image that has been saved in such a format in its past - or processed in any other way.

Part of the difficulty with performing high quality steganography and steganalysis is that most methods are heavily feature engineered, based partially on heuristics and semi-empirical rules. Therefore, finding a coherent theoretical way to describe steganography and steganalysis based on sound theoretical grounds is challenging, and while elegant theoretical concepts have been introduced to formalize the field many such formulations are of probably limited practical application [20]. One interesting frame of analysis, though, is the so-called 'model-based' approach [21], which establishes a clear link between the entropy of an image and the ability to hide information inside it, and proposes a method based on lossless encoders that could lead to practical methods.

2 A simple illustration of the vulnerability of LSB substitution

Simple Python code was developed to perform both LSB-substitution and LSB-matching on .png and .tiff images, and to test steganalysis algorithms. All the code used in this section is available on the Github of the author: https://github.com/jerabaul29/IllustrationSteganoSubstitutionVsMatching . The data to hide is randomly generated, so that it features as little structure as possible (similarly to what would be obtained encrypting the payload first). Tiff and png images are used, as they are either raw data, or compressed using lossless algorithms. The proportion of pixels in the images that are modified to hide data can be adjusted between 0 (no data hidden), and 1 (the LSBs of all pixels are used to hide data). The pixels used for hiding data are selected randomly in the whole image, so that the payload is randomly, statistically uniformly distributed in the image either on all three R, G and B channels or on a single one.

The steganalysis technique applied here against the LSB substitution method exploits the fact that LSB substitution breaks some symmetries in the histogram of the image. The histogram of an image is defined as the number of pixels having either their R, G or B channel taking each value between 0 and $2^N - 1$, with N the number of bits used to save each channel of the image (usually, $N = 8$):

$$H(n) = \#\{I(x,y) | I(x,y).R = n \text{ or } I(x,y).G = n \text{ or } I(x,y).B = n\}, \tag{1}$$

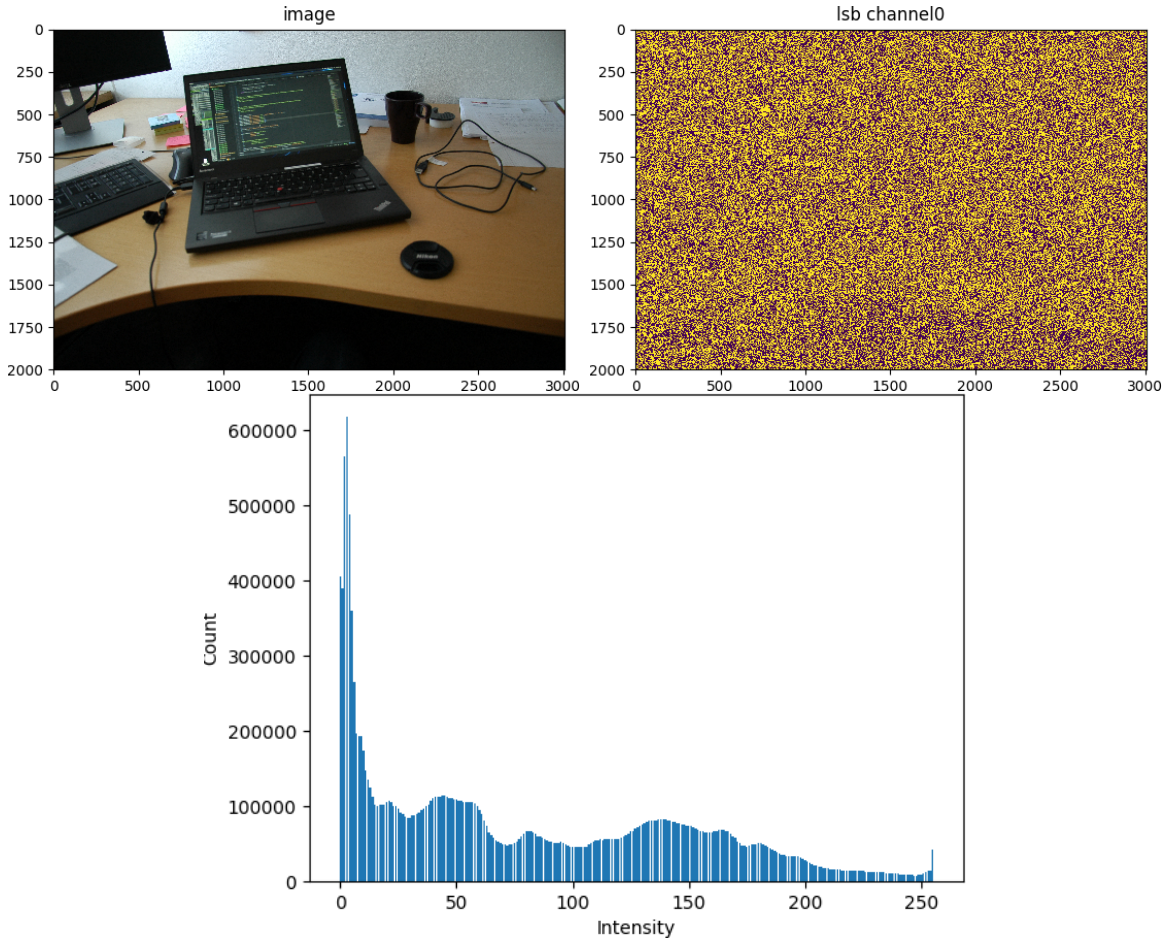

Figure 1: Top: digital image and LSB plane for R channel. Bottom: image histogram.

where H(n) is the value of the histogram computed on the image I, $0 \leq n \leq 2^N - 1$, # stands for "cardinal of ensemble" following, $I(x, y)$ is the pixel at location (x, y) of the image, and $I(x, y).R$ is the Red channel of the pixel (respectively Green, Blue indicated by G, B). Similar image histogram and steganalysis could also be applied on each channel separately. An example of picture, together with its histogram, is shown in Fig. 1.

One possible method in attacking LSB-substitution is based on observing that changes in pixel channel values introduced by LSB substitution correspond to diffusion between bins of the image histogram, but that only pairs of adjacent bins sharing all their N-1 bits above the LSB bit on the corresponding channel can exchange pixels during substitution. As a consequence, LSB-substitution will exchange pixels in the image histogram within intensity values pairs (0, 1), (2, 3), ..., $(2^N - 2, 2^N - 1)$, but not inside for example intensity pair (1, 2). Fig. 2 gives an illustration of this phenomenon. This is very similar in concept (but maybe simpler to understand) to the well-known RS steganalysis method and its derivatives: RS-steganalysis analyses the transition diagram for pixel pairs introduced by LSB-substitution.

There is no reason to expect that such grouping between adjacent pixels in the image histogram should, on average, occur on non-steganography images. Therefore, one can use this feature as a way to design a test giving the most likely proportion of pixels used for perform LSB-substitution. For this, the difference histogram is first computed:

$$D(n) = H(n + 1) - H(n), \tag{2}$$

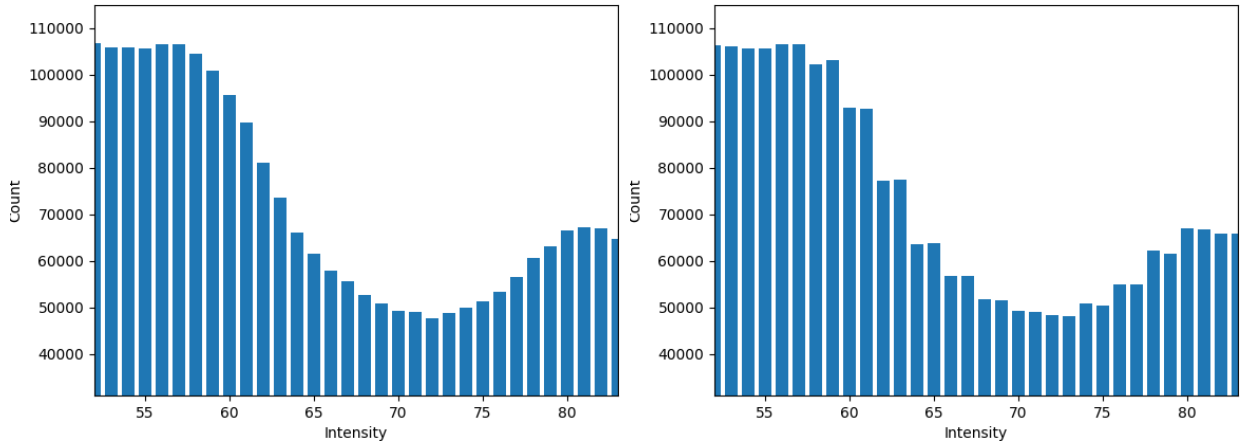

Figure 2: Left: image histogram when no LSB substitution has taken place. Right: image histogram after LSB substitution, embedding proportion of 1. The grouping effect of pairs of bins is clearly visible.

with $0 \leq n < 2^N - 2$. Then, the sum of the absolute values obtained in bins of even and uneven index of the difference histogram are computed (let us call them D_e for 'difference even', and D_u for 'difference uneven'). Bins of even index in the difference histogram indicate the difference of counts between bins of the image histogram that exchange pixels during LSB substitution, while bins of uneven index in the difference histogram indicate the difference of counts between bins in the image histogram that do not exchange pixels during LSB substitution. As a consequence, in the case without LSB substitution, we should expect $D_e \approx D_u$, while in the case with LSB substitution, $D_u = 0$. We then naturally adopt the following function for indicating the expected proportion of pixels used for steganography:

$$p_{predicted} = (D_e - D_u)/(D_e + D_u). \tag{3}$$

However, using this method directly on the difference histogram yields a lot of noise in the results. The reason for this is, some parts of the histogram will introduce noise in the method and should be discarded from the computation of D_e and D_u. For example, saturated bins at the extremities of the histogram, and bins located in area of the image histogram that are almost constant distort the results. To avoid this problem, we can select the bins to use in the computation of D_e and D_u using two different criteria: either a hard threshold (the value of the bins used must be comprised between a maximum and a minimum value), or a relative slope threshold. This last criterion works best. The idea is the following: the area where the distortion introduced by LSB substitution is most visible are those where, except for the effect of steganography, the histogram has an approximately constant slope. Selection of those regions in the image histogram is performed by comparing the first order finite difference (discrete derivative) and the second order finite difference (discrete second derivative) of the image histogram. Results of this selection process are illustrated in Fig. 3.

The performance of this steganalysis method, using both hard threshold and relative slope method for selection of relevant bins, is illustrated in Fig. 4. The images used for tests are a few shots taken by the author using a digital camera. While this is not, strictly speaking, an exhaustive test (one should perform tests on rather at least thousands of images to make such a claim), it illustrates the efficiency of the method. Fig 4 shows that the method works, which is striking owing to its simplicity, but it is possible that other algorithms such as RS-analysis could yield better results and better accuracy or detection threshold.

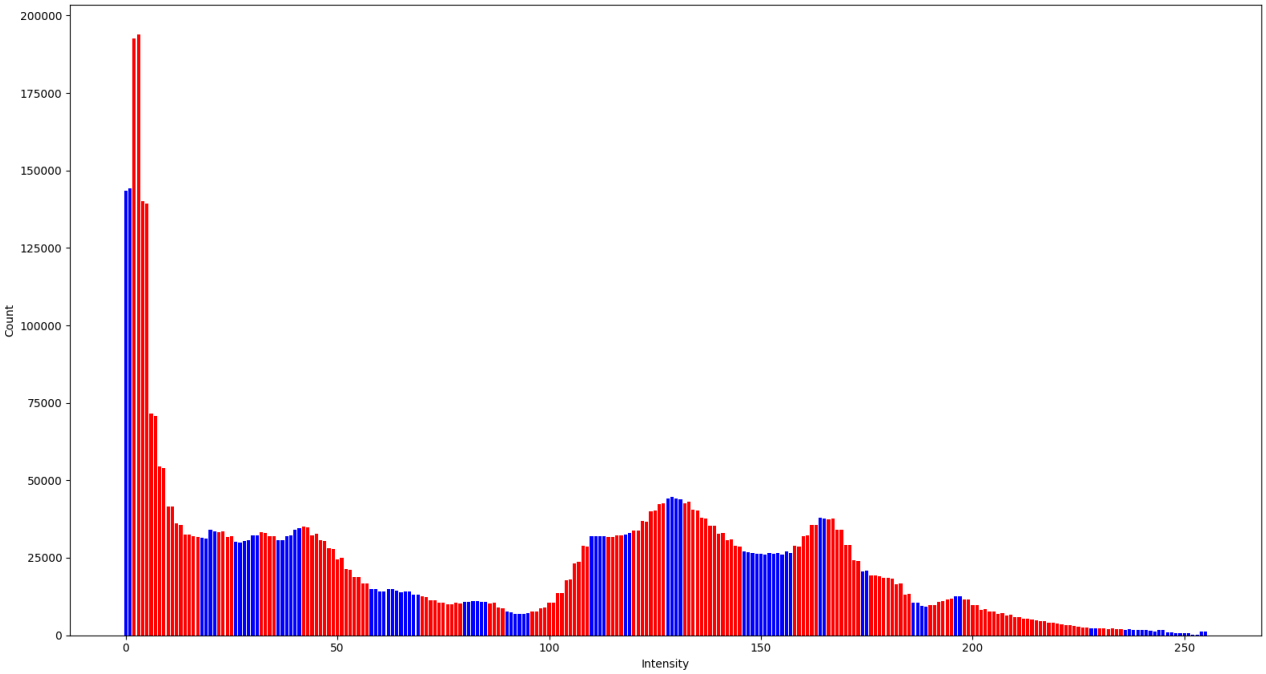

Figure 3: Illustration of the relative slope selection algorithm, applied on an image that has undergone LSB substitution with an embedding proportion of 1. Red bins are the ones that will be used for estimating the embedding proportion.

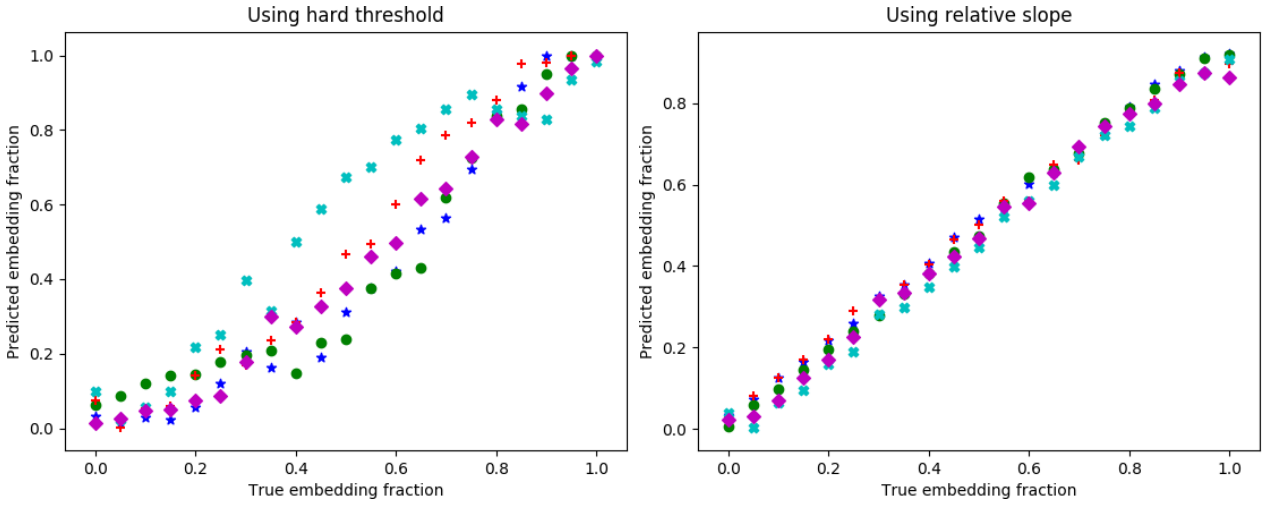

Figure 4: Illustration of the detection of LSB substitution, using the hard threshold (left) and the relative slope (right) selection criteria. Each symbol indicates a different image.

By contrast, LSB-matching does not introduce such an obvious symmetry breaking in the image histogram. When the value of a pixel channel must be changed to modify the LSB, it is either increased or decreased by one, both operations having a probability 1/2, therefore introducing diffusion in both direction in the image histogram. The average expected effect of LSB-substitution on a proportion p of the pixel channels on the image histogram can be computed as:

$$\begin{bmatrix} H'(0) \\ H'(1) \\ H'(2) \\ ... \\ H'(2^N-3) \\ H'(2^N-2) \\ H'(2^N-1) \end{bmatrix} = \begin{bmatrix} 1-p/2 & p/4 & 0 & ... & ... & ... & ... \\ p/2 & 1-p/2 & p/4 & 0 & ... & ... & ... \\ 0 & p/4 & 1-p/2 & p/4 & 0 & ... & ... \\ \vdots & 0 & \ddots & \ddots & \ddots & 0 & ... \\ \vdots & ... & 0 & 1-p/4 & 1-p/2 & p/4 & 0 \\ \vdots & ... & ... & 0 & 1-p/4 & 1-p/2 & p/2 \\ \vdots & ... & ... & ... & 0 & p/4 & 1-p/2 \end{bmatrix} . \begin{bmatrix} H(0) \\ H(1) \\ H(2) \\ ... \\ H(2^N-3) \\ H(2^N-2) \\ H(2^N-1) \end{bmatrix}, \tag{4}$$

where H' is the modified histogram. This transition matrix describes a diffusion problem, which tends to smooth the image histogram and is known to be difficult to inverse and will be difficult to detect a posteriori, as soon as the image histogram is expected to be approximately smooth already before steganography. By contrast, the transition matrix in the case of LSB substitution can be expressed as:

$$\begin{bmatrix} H'(0) \\ H'(1) \\ H'(2) \\ H'(3) \\ ... \end{bmatrix} = \begin{bmatrix} 1-p/2 & p/2 & 0 & ... & ... & ... \\ p/2 & 1-p/2 & 0 & 0 & ... & ... \\ 0 & 0 & 1-p/2 & p/2 & 0 & ... \\ 0 & 0 & p/2 & 1-p/2 & 0 & ... \\ ..... \end{bmatrix} \cdot \begin{bmatrix} H(0) \\ H(1) \\ H(2) \\ H(3) \\ ... \end{bmatrix}, \tag{5}$$

where the symmetry breaking corresponding to count exchanges only between specific pairs is clearly visible. As a consequence of this difference between LSB substitution and LSB matching, LSB matching is obviously immune to the analysis technique used against LSB substitution, as illustrated in Fig. 5.

Note, however, that the transition matrix for LSB matching is disturbed near the edges of the image histogram. Therefore, one could possibly use this property to design some LSB matching detection tests, and one should avoid to perform LSB matching on images presenting some saturated regions (i.e., images with empty extremal bins should be preferred).

What is underlined here is the fact that, while LSB substitution is 'obviously' vulnerable to simple attacks based on the modifications it introduces in the image histogram, such an obvious flaw is not present in the same form with LSB matching. While this proves that LSB substitution is compromised this does not prove that LSB matching is safe. Actually, several methods have been reported to be successful at attacking LSB matching (even if those methods are arguably more complicated and less reliable than those that can reveal LSB substitution). One should also be aware of the fact that image histograms considerably reduce the amount of information initially present in the image, and therefore more powerful techniques for detection of LSB substitution, that take advantage of more of the information available in the image, are likely to exist (actually, RS-steganalysis is one such method).

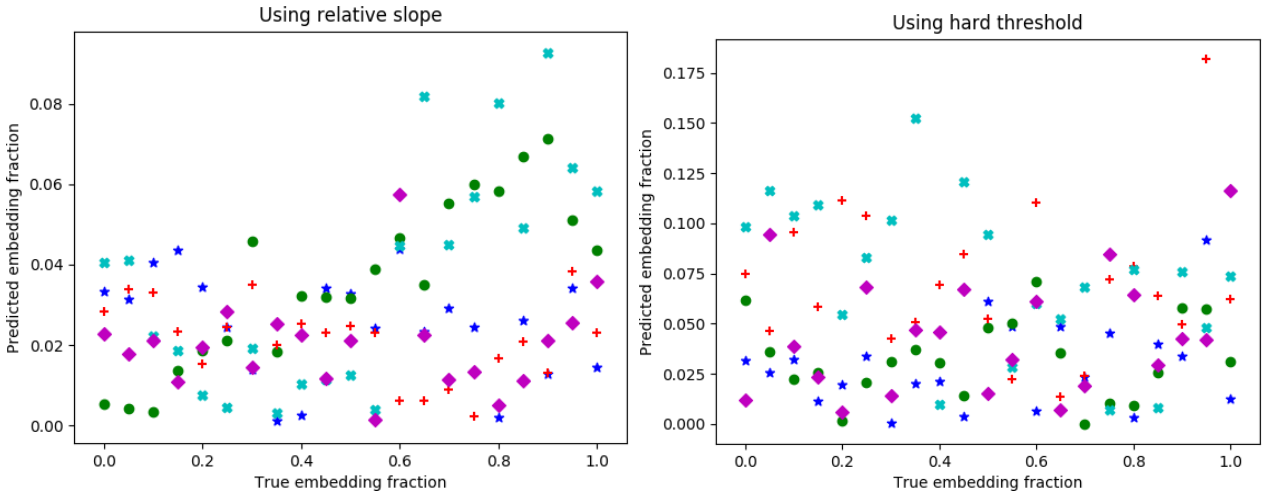

Figure 5: Illustration of the fact that, obviously, the method previously presented for detecting LSB substitution does not work against LSB matching. Left: using the hard threshold, and right: using the relative slope selection criteria. Each symbol indicates a different image.

3 Conclusion

Performing undetectable steganography is challenging. Firstly, one needs to find features in the image that are noise, and can be used for hiding information. This is not an easy task, as one cannot prove that some data in an image is noise if not random noise is actively injected first - it is always possible that there exists an explaning factor, unknow to the steganograph but known to the steganalyst. It has been suggested that a way to avoid this issue could be to actively add noise to the images before steganography is performed, but this approach is probably not valid. Indeed, it is equivalent to perform steganography directly, using encryption first (as 'perfect' encryption will transform the message into a pseudo-random string indistinguishable from random for anyone but those possessing the encryption key). One could argue, though, that images could be turned into a 'perfect' steganography cover by systematically adding such a random noise, independently of the will of the user to actually perform steganography, but then either few actors will apply this principle - and raise suspiscion upon them -, or if every single image coming out of every camera sensor was to be processed in such a way - for example, by making such addition of noise mandatory on all cameras -, this would represent a large waste of storing capability therefore only applicable through constraint on the standard user, who is not interested in steganography.

Secondly, even if some random component is present in the image that can be used for steganography, as it is probably the case in reality (LSBs are very likely at least partly random because of sources of noise in the sensor that arise from physical, 'truly' random processes such as thermal noise and quantum fluctuations), it is difficult to alter this random data without letting evidence of the alteration. This is because, while random, the noise used to hide information is still correlated in a non trivial way with the rest of the information in the image, and therefore some structures may still be present in this randomness. In the case of LSB substitution, one sees through the image histogram that while every single pixel LSB is probably random, the probability of having a 0 or a 1 is different and correlated with the local slope of the image diagram. The difference in probability between a LSB 0 or 1 is especially important in regions where the image histogram has strong, approximately constant gradient, which allows to build reliable and accurate detection tests.

This discussion underlines the fact that, ultimately, one needs a comprehensive (and accurate) model of the noise present in an image in order to perform steganography in a way that does not alter the support in a distinguishable way. In this context model-based steganography appears as a promising framework for understanding, analyzing and developing steganography and steganalysis techniques. Using this framework of analysis, the model based on the assumption that the LSB value taken alone is random (and can be substituted by a pseudo-random encrypted message) is grossly invalid as it completely ignores relation with higher order bits, which is the reason for the failure of LSB substitution. By contrast the model based on the assumption that the value of a pixel has an uncertainty in the range, at least, of ± 1, is sounder as it naturally involves primarily the LSB value, but also by cascade higher order bits (in a 'naturally' decreasing extend as the bit considered is of higher order), and preserves some structures naturally present that were disturbed by LSB substitution. However, formulated in this way, LSB matching also has obvious shortcuts: probably the uncertainty around the value of a pixel is dependent on a number of parameters such as local variance in the image, the presence of an edge in the image close to the pixel considered, etc. Ignoring those features in the steganography process could possibly leave a signature on the modified image.

One should therefore be very careful about the steganography method used, and the support images selected. While too simplistic models such as the one underlying LSB substitution lead to easy detection of data embedding, too sophisticated methods are probably also dangerous to use as their complexity means that they may introduce signatures that the steganographist may not be aware of. Similarly, the use of images that have been compressed, or processed in any other way, is a bad idea as such preliminary alteration introduces even more structures in the image that are possible vulnerabilities exploitable for steganalysis. For example if you find an image that is nearly exactly the result of jpeg compression, because it presents typical jpeg artifacts, but cannot exactly be encoded following the jpeg standard, it is very likely that it was first compressed, and then altered. Actually, the difficulty of not distorting any structure in an image has motivated several authors to recommend the use of grayscale images that have never undergone compression or processing to hide information. It is also well known, for the reason previously underlined, that gif, jpeg and other lossly encoded or computer generated images should not be used for any kind of LSB steganography. Of course, the relative percentage of changes introduced by steganography is a key parameter in the risk of detection. It is well documented that one should try to limit the proportion of pixels affected by steganography, and hence the embedding fraction, to limit detectability ([3] advises not to use a number of embedding slots higher than $\sqrt{n}$, where n is the total number of available slots). This comes at the cost that the payload needs to be split between several images. If a large payload needs to be split between several images in a folder, the same rule of limited distortion may apply to the number of images used for steganography: one should probably use only a limited fraction of the images as steganographic covers, and let a number of them unaffected.

As mentioned previously, the model based steganography approach is -to the knowledge of the author- the only sound theoretical frame for building a theory of steganography, and could be a satisfying solution to the race between highly specialized, feature-engineered steganography and steganalysis methods that are regularly appearing. What the model-based approach implies is that, unsurprisingly, the sword and shield race reduces to a battle for getting the best model of an image, which could be used both to hide (steganography), and to measure (steganalysis) image distortions.

As a last comment, being successful at hiding information implies that one should be careful about not only the steganography process itself, but also about not letting other evidences behind. A chain is only as strong as its weakest link, and issues such as uncleared logs (including

terminal logs if a command line tool is used, or software logs if steganography is performed with a software), compromised computers, or keeping the original picture instead of deleting it (so that one can trivially detect modifications to the image), could compromise even 'perfectly secure' steganography.

References

[1] Spammimic - hide a message in spam; url: http://www.spammimic.com/, accessed 07/2017.

[2] T Morkel, JHP Eloff, and MS Olivier. An overview of image steganography.

[3] Andrew D. Ker, Patrick Bas, Rainer Böhme, Rémi Cogranne, Scott Craver, Tomáš Filler, Jessica Fridrich, and Tomáš Pevný. Moving steganography and steganalysis from the laboratory into the real world. In *Proceedings of the First ACM Workshop on Information Hiding and Multimedia Security*, IH&MMSec '13, pages 45–58, New York, NY, USA, 2013. ACM.

[4] Wen chen. Study of Steganalysis methods. Master's thesis, New Jersey's Science Technology University, USA, 2005.

[5] S. Dumitrescu, Xiaolin Wu, and Zhe Wang. Detection of lsb steganography via sample pair analysis. *IEEE Transactions on Signal Processing*, 51(7):1995–2007, July 2003.

[6] Andrew D. Ker. *A General Framework for Structural Steganalysis of LSB Replacement*, pages 296–311. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[7] S. Dumitrescu, Xiaolin Wu, and N. Memon. On steganalysis of random lsb embedding in continuous-tone images. In *Proceedings. International Conference on Image Processing*, volume 3, pages 641–644 vol.3, June 2002.

[8] Andrew D Ker. Resampling and the detection of lsb matching in color bitmaps. 2005.

[9] A. D. Ker. Steganalysis of lsb matching in grayscale images. *IEEE Signal Processing Letters*, 12(6):441–444, June 2005.

[10] G. Cancelli, G. Doerr, I. J. Cox, and M. Barni. Detection of x00b1;1 lsb steganography based on the amplitude of histogram local extrema. In *2008 15th IEEE International Conference on Image Processing*, pages 1288–1291, Oct 2008.

[11] Jessica Fridrich, David Soukal, and Miroslav Goljan. Maximum likelihood estimation of length of secret message embedded using ±k steganography in spatial domain, 2005.

[12] F. Huang, B. Li, and J. Huang. Attack lsb matching steganography by counting alteration rate of the number of neighbourhood gray levels. In *2007 IEEE International Conference on Image Processing*, volume 1, pages I – 401–I – 404, Sept 2007.

[13] Zhihua Xia, Xinhui Wang, Xingming Sun, Quansheng Liu, and Naixue Xiong. Steganalysis of lsb matching using differences between nonadjacent pixels. *Multimedia Tools and Applications*, 75(4):1947–1962, Feb 2016.

[14] J. Mielikainen. Lsb matching revisited. *IEEE Signal Processing Letters*, 13(5):285–287, May 2006.

[15] X. Li, B. Yang, D. Cheng, and T. Zeng. A generalization of lsb matching. *IEEE Signal Processing Letters*, 16(2):69–72, Feb 2009.

[16] Li Fan, Tiegang Gao, Qunting Yang, and Yanjun Cao. An extended matrix encoding algorithm for steganography of high embedding efficiency. *Computers Electrical Engineering*, 37(6):973 – 981, 2011.

[17] Andreas Westfeld. *F5—A Steganographic Algorithm*, pages 289–302. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[18] Jessica Fridrich, Miroslav Goljan, and Dorin Hogea. *Steganalysis of JPEG Images: Breaking the F5 Algorithm*, pages 310–323. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

[19] Rainer Böhme and Andreas Westfeld. *Breaking Cauchy Model-Based JPEG Steganography with First Order Statistics*, pages 125–140. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[20] Christian Cachin. An information-theoretic model for steganography. *Information and Computation*, 192(1):41 – 56, 2004.

[21] Phil Sallee. *Model-Based Steganography*, pages 154–167. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.