Steganography techniques

Dragoş Dumitrescu[1], Ioan-Mihail Stan[1], Emil Simion[2]

[1]University Politehnica of Bucharest, Faculty of Automatic Control and Computers, Computer Science Department
[2]University Politehnica of Bucharest, Faculty of Applied Sciences, Department of Mathematical Models and Methods
dragos.dumitrescu92@stud.acs.upb.ro, ioan.stan@stud.acs.upb.ro, emil.simion@upb.ro

Abstract. As cryptography is the standard way of ensuring privacy, integrity and confidentiality of a public channel, steganography steps in to provide even stronger assumptions. Thus, in the case of cryptology, an attacker cannot obtain information about the payload while inspecting its encrypted content. In the case of steganography, one cannot prove the existence of the covert communication itself. The main purpose of the current paper is to provide insights into some of the existing techniques in steganography. A comparison between the performances of several steganography algorithms is accomplished, with focus on the metrics that characterize a steganography technique.

Keywords: steganography, information hiding, covert communication, privacy

1 Introduction

With the strong development of computing, large amounts of media are constantly being downloaded and streamed across the internet. The variety of these media leads to difficulties in analyzing normal and abnormal content within. Also, as most processes in the Internet are driven by humans, predicting behavior and analyzing anomalies is a complicated process that may require high computing power and sophisticated algorithms.

Steganography relies on this unpredictability in order to perform information hiding inside apparently innocuous payloads. While in the case of cryptography the main focus resides in the attacker not being able to get information on the payload from its encrypted content, steganography aims at creating a communication channel between two parties, without an intermediary noticing the existence of the particular channel. One can easily conclude that the assumptions offered by steganography are stronger than those offered by cryptography.

As cryptanalysis is the counterpart of cryptography, steganalysis is the counterpart of steganography. A *steganalyst* tries to determine the existence of a covert communication channel between two parties and either break or alter their communication. While cryptology states that a cipher is broken when the attacker is able to gain information on the content of the payload, a steganography technique is considered broken when its mere existence is proven.

The art of information hiding was first accounted for in the work *Histories* by Herodotus around 440 B.C, where he describes a technique to carry secret messages by imprinting the secret message on the shaved head of a slave. Upon hair growth, the mere presence of the message was unknown to an enemy [1]. The etymology of the term *steganography* is Greek and derives from *steganos* – hidden and *graphein* – writing.

In the next section, theoretical insights into the field of steganography are given with an information theoretic approach, emphasizing on the metrics that a steganography algorithm is characterized by. In the third section, several steganography techniques are described as references for the envisaged tests to be performed. In the fifth section, the proposed steganography test suite to be employed is described, with focus on the rationale behind each test included.

2 Background

The current section focuses on providing the information theoretical background on steganography and includes mathematical definitions and theorems. In the second part, the metrics used to characterize the performance of a steganography algorithm are depicted and means to calculate them are described.

The seminal papers for the following section are Cox's *Digital watermarking and steganography* [2], chapters 12 and 13 and Kaltzenbeisser's and Petitcolas's *Information Hiding techniques for steganography and digital watermarking* [3], chapters 1-4.

In one of his seminar papers in secrecy systems, Shannon stated that systems for hiding information are *"primarily a psychological problem"* and did not undergo a rigorous theoretical approach on the topic [4].

Formulating the steganography problem is due to Simmons [5]. Accordingly, the problem of steganography is that of two prisoners, Alice and Bob who are trying to exchange messages while being constantly intercepted by the prison's warden Wendy. Should Wendy consider the messages exchanged between Alice and Bob suspicious, she will drop their communication. The above model applies in reality, for instance under oppressive regimes or under governmental policies that disable the use of cryptography within the boundaries of a particular country. Thus, the drive for confidentiality between two parties creates the need for information hiding schemes, such that the warden cannot make the difference between a secret message transmitted among the parties and a regular message as part of their conversation.

In the following paragraphs, we present the theoretical model behind steganography as described by Cachin in [6]. In [2] and [3], the same theoretical definitions are presented.

2.1 Theoretical model

Preliminary to defining the steganography system, let Alice and Bob be possessors of a shared *secret key*, only known between one another. Their purpose is to communicate *secret messages* to one another, while only being able to send messages from a given set of *covers.*

In Fig. **1**, a general block scheme diagram represents the main components of a *stego-system*. Note that the diagram only takes into account the *passive attacker* assumption. In this approach, the attacker – Wendy – is bound not to modify the contents of the *Stego Object*, the only action she can perform is drop or allow the passage of the message. Should the warden be able to alter the payload of the stego object, then the attacker is called *active*.

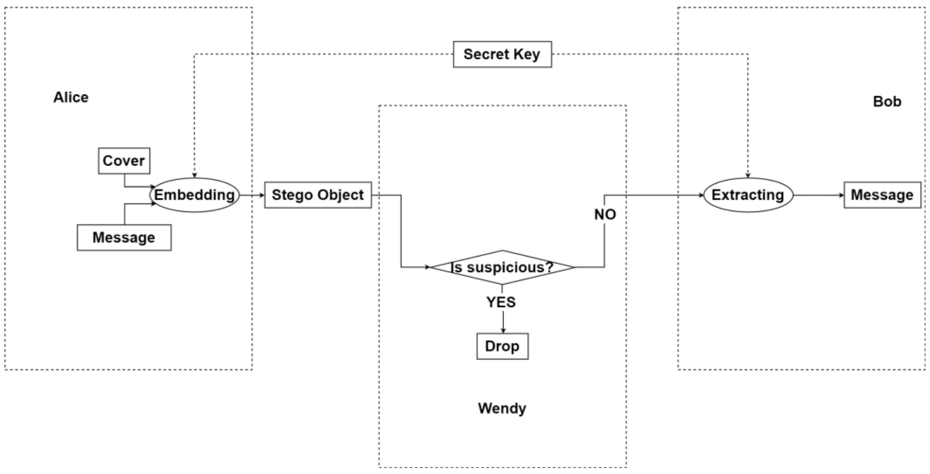

Fig. 1. General block scheme of a stego-system

Definition. A steganography system is a quintuple $\wp = (C, M, K, D_K, E_K)$, where C is the set of all covers used in communication, M is the set of all secret messages that need to be transported using the covers, K the set of secret keys, $E_K: C \times M \times K \to C$, and $D_K: C \times K \to M$ two functions, the embedding and the extraction functions respectively such that: $D_K(E_K(\boldsymbol{c}, \boldsymbol{m}, \boldsymbol{k}), \boldsymbol{k}) = \boldsymbol{m}.$

Note that in the definition above, no care is taken in what concerns the means by which Alice and Bob handle the key exchange. Under the assumption of an existing shared secret key between the two parties, the framework discussed above is named *secret key steganography*. Its counterpart, *public key steganography* is based on the same principle as *public key cryptography* (for further details, see [3]). Another category of steganography techniques is that of *pure steganography* [3]. Pure steganography does not assume the existence of a shared secret between the two parties. In fact, the effectiveness of a pure stego-system lies in the secrecy of the two embedding functions, thus violating Kerchoff's principles – the security of the system should only depend on the secrecy of the key and not on that of the algorithm. In the current paper, the focus will lie solely in shared key steganography and its applications.

Let P_C be a probability distribution over the set of all covers and let P_S be a probability distribution over the stego-objects. The relative entropy or the Kullback-Leibler distance between two probability distributions P_C and P_S is given by:

$$D(P_C \parallel P_S) = \sum_{c \in C} P_C(\boldsymbol{c}) \log\left(\frac{P_C(\boldsymbol{c})}{P_S(\boldsymbol{c})}\right)$$

Note that the above relation is not an actual distance in the geometrical sense – since it is not symmetrical and does not obey the triangle's inequality. However, quantifying how "different" two probability distributions are can be easily performed using this metric. Note that, if $P_C = P_S$, then the distance is 0, which is an intuitive result.

Definition. Let $\wp$ be a stego-system and let P_C and P_S be the two probability distributions of cover messages and stego objects. $\wp$ is called ϵ-secure against passive attackers if:

$$D(P_C \parallel P_S) \leq \epsilon$$

If $\epsilon = 0$, then the stego-system is called perfectly secure.

Proposition. There exists a perfectly secure stego-system.

Proof of the above proposition is not included in this paper. However, note that the construction of a *perfectly secure* stego-system is analogous to that of constructing a perfectly secure encryption system as it employs the *one-time pad* embedding process in order to guarantee exact distribution of the stego-objects in relation to that of cover objects. This kind of system is, however, only theoretically feasible since the key must equal the message size and also one must never reuse the same key for covering different messages.

The actions of Wendy can be thought of as hypothesis testing, where:

H_0: The message does not contain a secret message

H_1: The message contains a secret message

A *false positive* is a decision whereby Wendy decides to block an innocent message. A *false negative* is a decision whereby Wendy allows passage of a cover containing a secret message.

Theorem (Cachin). Let $\wp$ be an ϵ-secure stego-system and let α the probability of a false positive and β the probability of a false negative. Let:

$$d(\alpha, \beta) = \alpha \log_2 \frac{\alpha}{1-\beta} + (1-\alpha) \log_2 \frac{1-\alpha}{\beta}$$

Then:

$$d(\alpha, \beta) \leq \epsilon$$

It is not uncommon to consider $\alpha = 0$ (i.e. Wendy cannot make false *accusations*). In this case, $\beta \geq 2^{-\epsilon}$.

The proof [6] of the above result is beyond the scope of this paper; intuitively, for $\alpha = 0$, it is easy to see that as ϵ decreases, it is exponentially less likely for Wendy to drop payloads that actually contain secret messages.

Another aspect that needs to be taken into account when designing a steganography algorithm is that whatever alterations are performed on the target object, the distortion between the initial cover and the final stego-object needs to be minimal. This assertion is due to the fact that a human or automated warden might notice artifacts introduced within the transmitted payload.

Definition [3]**.** A function on a set C is called *similarity* if $sim: C \times C \to (-\infty, 1]$ with:

$sim(x, y) = 1 \Leftrightarrow x = y$ and $sim(x, y) < 1, \forall\, x, y \in C$ and $x \neq y$

For a practical stego-system, one usually picks E_K such that $sim(E_K(\boldsymbol{c}, \boldsymbol{m}, \boldsymbol{k}), \boldsymbol{c}) \approx 1, \forall \boldsymbol{c} \in C, \boldsymbol{m} \in M, \boldsymbol{k} \in K$.

Depending on the nature of C, one can identify several media into which secret messages can be embedded. The most common in digital steganography are:

- Image-based techniques – the carrier is an image. Usually, one hides messages in the noise component of a given image.
- Sound-based techniques – the carrier is sound.
- Text-based techniques – can consist of typos, spacing schemas, rendering mistakes et.al. in order to convey messages
- Network packets – hiding is performed in the unused headers of an TCP/IP packet
- OS-hiding – embedding in unused portions of the hard-drive

Depending on the nature of the communication method, one can also distinguish between random-access media – such as sound files, images, movies et.al. – where the user has access to all information comprised within the image and streaming media, in which case the user has access to just a portion of the full message.

Depending on the mutation space of the embedding algorithm, there are methods altering the cover in its direct space (time in audio sounds, pixel number in images), in the transform space (DCT domain, Fourier domain et.al.).

Several algorithms aim at preserving some theoretical model of the cover, while others intend to preserve overall image statistics.

In the following paragraphs, important metrics for characterizing steganographic algorithms are listed.

Let $l: C \to \mathbb{N}$ be a function representing the length of a given cover. Likewise, $l: M \to \mathbb{N}$ is the length of a given embeddable message.

The *embedding capacity* of a stego-system is given by $\log_2 |M|$ and its unit is bits. Let m be the length of a message. If m is smaller than the embedding capacity, the *relative length* is given by $\frac{m}{|M|}$.

Let $D: C \times C \to \mathbb{R}_+$ be a distance defined on the cover space. Such a distance could be the mean square error, or the Hamming distance. The average number of bits embedded per unit distortion is called the *embedding efficiency*.

Note that the *embedding efficiency* clearly characterizes a stego-system. The higher the embedding efficiency, the more bits one can embed without producing major distortions to the initial payload, thus making the attacker less likely to detect tampering with the "normal" covers.

All of the above metrics to characterize a stego-system rely on the assumption of a *passive attacker*. In case Wendy willingly distorts the contents of the payloads transmitted between Alice and Bob, she is called an *active attacker*. If Wendy tries to inoculate new semantics to the two prisoners' covert communication, she is called a *malicious attacker.*

A reasonable metric for a stego-system to withstand tampering by an attacker is *robustness*.

Definition. Let $\wp$ be a stego-system and let $\mathcal{H}$ be a set of mappings $h: C \to C$. $\wp$ is *robust* to $\mathcal{H}$ iff:

$$\forall h \in \mathcal{H}, m \in M, c \in C \text{ and } k \in K, \text{then } D_K\big(h\big(E_K(\boldsymbol{c}, \boldsymbol{m}, \boldsymbol{k})\big), \boldsymbol{k}\big) = \boldsymbol{m}$$

The trade-off between assuring robustness and the *embedding capacity* is obvious, since redundant information about the embedded message needs to be also included in the payload, thus reducing the amount of *useful* information to be sent.

As can be clearly observed in literature, the most spread steganography techniques are image-based. In the current paper, images will be described as a 2D matrix of real or integer values, where each row and column represent an x and y pixel respectively. Commonly employed techniques include: LSB encoding, DCT-based encoding and spread-spectrum techniques. The former has the advantage of computational efficiency as well as simplicity, but this can be show to violate *regular* image statistics. Several techniques derived from LSB embedding, are depicted in [7].

3 Literature survey

In the following paragraphs, the main interest lies in describing several well-known steganography techniques, as well as their drawbacks. The methods described herein concern image-based steganography and attempt to cover the major approaches for this particular category: direct space techniques, transform space techniques and spread-spectrum techniques. Each technique described within the current paper will be submitted to a test suite, in order to describe the amount of distortion induced as a function of the size of the payload. Thus, the following metrics will be employed:

— Mean Squared Error (MSE):

$$MSE(A, \mathrm{B}) = \frac{1}{n}\sum_{(i,j)} (A[i,j] - B[i,j])^2 \quad (1)$$

— Peak Signal to Noise Ratio (PSNR):

$$PSNR(A, B) = 10 \cdot \log_{10} \frac{MAX_I}{MSE(A,B)} \; (2),$$

Where MAX_I is the value of the brightest pixel.

Mean Structural Similarity (MSSIM). Since its main purpose is to focus mostly on structure rather than average power levels regarding a particular image, this metric is often seen as superior to the first two, since it conveys better human visual system response to stimuli. Thus:

$$SSIM(x,y) = \frac{(2\sigma_{xy} + C_2) \cdot (2\mu_x\mu_y + C_1)}{(\sigma_x^2 + \sigma_y^2 + C_2) \cdot (\mu_x^2 + \mu_y^2 + C_1)} \quad (3)$$

Where:
The signals x and y are corresponding windows inside the original image.
$\mu_x = \sum_{i=1}^{N} x_i$ is the mean of the signal x
$\sigma_x = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \mu_x)^2}{N-1}}$ is the standard deviation of the signal x
$\sigma_{xy} = \frac{\sum_{i=1}^{N}(x_i - \mu_x)(y_i - \mu_y)}{N-1}$ is the covariance of the two signals x and y
C_1, C_2 are two constants taken for stability purposes in the case where either term of the denominator is close to zero.

A classification of image steganography techniques is provided by Johnson and Kaltzenbeisser in [3]. Thus, they identify six categories of steganography,

- Substitution systems
- Transform domain techniques
- Spread spectrum techniques
- Statistical methods
- Distortion techniques
- Cover generation methods

The following paragraphs shall focus mostly on the first three of the above and will provide an example for each of them. Thus, the first paragraph will cover the *Least Significant Bit* technique; the second will cover the *F5* algorithm, while the third will focus on the *SSIS* – spread spectrum image steganography – technique.

3.1 LSB

The last significant bit method is one of the most efficient and popular mechanisms to embed a secret message into a photo. This one is based on the fact that human’s visual perception may not precisely detect the subtle alteration of the hue of a specific color.

An image is represented as a matrix of pixels. Each pixel is composed by at least one channel representing the intensity of a color on that particular segment of image. The most common way to represent a digital photo is by using the RGB model, which enables 3 different channels per pixel. The last significant bit method observes and exploits the fact that increasing the intensity of the color components with one or two units, the change will not be perceived by the human eye.

Fig. 2, Fig. 3 and Fig. 4 reveal how a small adjustment of each color component is not changing the overall impression for even the finest observer.

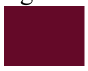

Fig. 2. R=100, G=10, B=40 **Fig. 3.** R=101, G=11, B=41 **Fig. 4.** R=103, G=13, B=43

In this respect, the secret information can be embedded, bit by bit, into the last significant bit of each color component. The color predominance factor is represented in bytes, thus, if the secret message is an image, the carrier image must be at least 8 times bigger than the embedded one, in order to reduce the risk to be discovered. However, in practice, altering the last two or three significant bits may not change the visual impression (Fig. 4). In this respect, the carrier image can be smaller.

A QR code (Quick Response code) is a digital image that is able to carry up to 4296 alphanumeric characters. This is a form of bar coding which is used widely as label for different items in order to maintain as many information as possible in a small, compressed machine-readable optical matrix. For our further example, a QR Code (Fig. 5) will be hidden in a landscaping image (Fig. 6).

Fig. 5. QR Code Gray Scale
150 x 150
containing "This is my secret text!" message

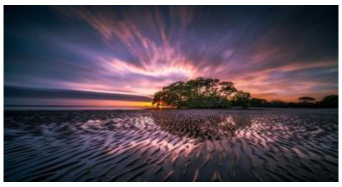
Fig. 6. Original RGB Image 1280x850

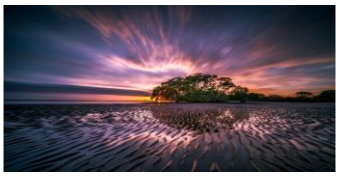
Fig. 7. LSB Method RGB Image 1280x850

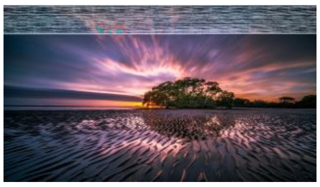
Fig. 8. MSB Method RGB Image 1280x850

By combining Fig. 5 and the carrier image Fig. 6, the result (Fig. 7) is similar to the original.

To see how the photo was changed, instead of changing the last significant bit, the most significant bit will carry the hidden information (Fig. 8).

After extracting the secret message from Fig. 7, one obtains **Fig. 9**.

Fig. 9. Secret Message from **Fig. 5** - QR Code 150x150

The LSB method is not limited to use only photos as secret messages. The mechanism is compatible with any form of digital information that can be comprised in the dimension constraints of the carrier photo. In this regard, instead of embedding human readable information such as photos containing text, facts, landscapes or portraits, clear text documents, audio segments and other perceptible items, an encryption algorithm can be applied before using the LSB processor.

Several considerations must be taken into account when dealing with the LSB coding technique. Let $l(c)$ the length of the cover image given by $l(c) = W \cdot H$ and also $l(m)$ be the length of the message to be embedded. The question that arises is where Alice places the $l(m)$ bits. A simple approach is to choose indices$\{j_i | i = 1..l(m)\} \subset \{1..l(c)\}$ in the left-right top-down order. A different approach could be that, starting from the stego-key K, Alice can generate a pseudo-random function between the set $\{1..l(m)\}$ and the set of $\{1..l(c)\}$ such that the distribution of the message bits throughout the cover is more uniform. This approach, however, incurs penalties in terms of computational complexity, since it requires checking for possible collisions in the function generation and may also include computing hash functions. Another difficulty in the LSB approach resides in that Bob has no knowledge of the length of the message prior to decoding it. Therefore, an encoding mechanism needs to be established between the two parties.

Fig. 10. Original RGB Image 1280x850

Fig. 11. Original RGB Image 640x397

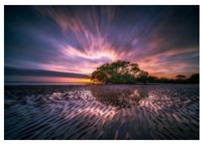
Fig. 12. LSB Method RGB Image 1280x850

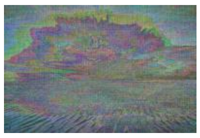
Fig. 13. MSB Method RGB Image 1280x850

For breaking the classic LSB encoding scheme we employ what is known as the Laplace filtered image:

$$\Delta p(x,y) = p(x+1,y) + p(x-1,y) + p(x,y+1) + p(x,y-1) - 4p(x,y)$$

The histogram of the Laplace filtered image on any natural image is, as expected tightly clustered around 0. Nevertheless, changing the LSB of each pixel in the image will eventually modify the distribution of $\Delta p(x,y)$ and increase the width of its peak around 0. Even though this metric is not a definitive proof of tampering with the original image, it represents strong evidence of tampering. In **Fig. 14**, histograms of the Laplace transform of an untampered image and of some LSB embedded images are presented. Notice how the central lobe diminishes in height, but gains in width as the message length increases.

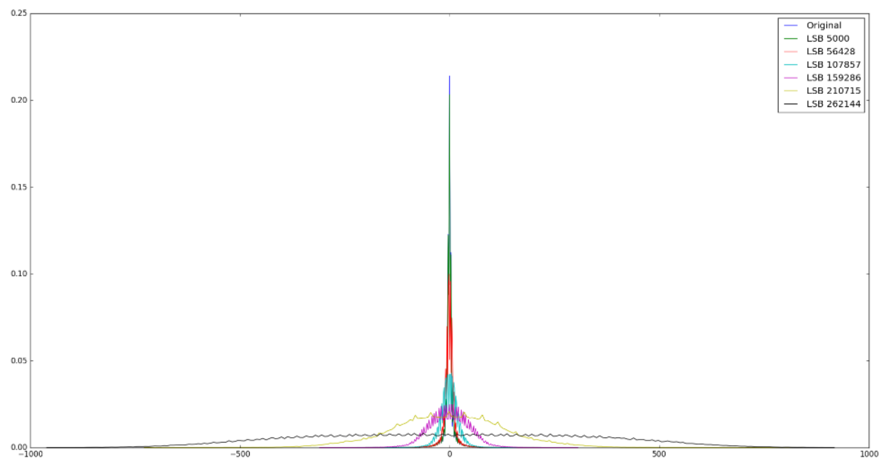

Fig. 14. Histograms of Laplace transformed images (original and LSB embedded with variable payload size)

3.2 F5

The simplicity of the LSB family of methods makes them popular choices for steganography. Nevertheless, they perform relatively poorly in preserving statistical measures, under the assumption of a uniform distribution in the message space. Another disadvantage of the aforementioned lies in their lack of robustness even to the simplest kinds of image manipulation – e.g. JPEG compression.

In this paragraph, the approach towards transform-space based embedding schemas is depicted, with focus on the F5 algorithm [7]. The preliminaries will describe a method similar to that proposed by Zhao in [8] and explained in [3].

Let $I(x, y)$ be an image with $x, y \in \{0..N-1\}$ a pixel position in a $N \times N$ square block. The DCT transform of an image is a $N \times N$ matrix of numbers given by:

$$DCT\{I\}(x,y) = \frac{1}{\sqrt{2N}} C(x) \cdot C(y) \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} I(i,j) \cdot \cos\left(\frac{(2i+1)\cdot x\pi}{2N}\right) \cos\left(\frac{(2j+1)\cdot y\pi}{2N}\right) \quad (4)$$

Where,

$$C(x) = \begin{cases} \frac{1}{\sqrt{2}}, x = 0 \\ 1, x > 0 \end{cases}$$

The DCT transform is crucial for the JPEG lossy compression method. In **Fig. 15**

, a simplified version of the JPEG compression scheme is presented. The image is first split into multiple 8×8 blocks. Each block is DCT-transformed and then quantized using some quantization table. Quantization is the actual lossy step in the compression scheme since, by definition:

$$Q(i,j) = \frac{I(i,j)}{Table(i,j)} \text{ rounded to the nearest integer}$$

Notice that quantization tables have increasing coefficients corresponding to higher order frequency terms. The reason for this resides in the way natural images are composed. For a natural image, the highest frequency component is the DC component (that is $x = y = 0$) and gets less significant as the frequency increases.

The lossless compression step includes different techniques such as *Run Length Encoding*, *Entropy Coding* (Huffman compression) et.al. Since these details are not part of the steganography methods described herein, they will be omitted for brevity.

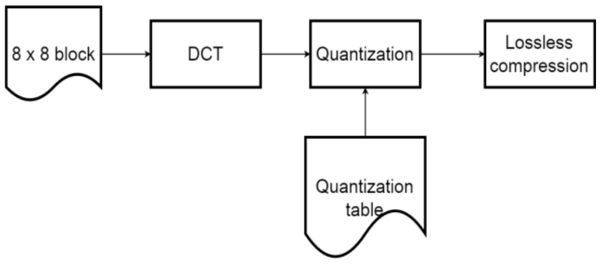

Fig. 15. JPEG simplified workflow

The steganography technique proposed by Zhao and Koch [8] operates on the quantized DCT coefficients. It employs three shared pairs of coefficients in the DCT block, namely $(u_1, v_1), (u_2, v_2), (u_3, v_3)$. Then:

if $B_i^Q(u_1, v_1) > B_i^Q(u_3, v_3) + D$ and $B_i^Q(u_2, v_2) > B_i^Q(u_3, v_3) + D \Rightarrow B_i$ encodes 1

if $B_i^Q(u_1, v_1) + D < B_i^Q(u_3, v_3)$ and $B_i^Q(u_2, v_2) + D < B_i^Q(u_3, v_3) \Rightarrow B_i$ encodes 0

Following the above assumptions, Alice needs to change the values of the respective quantized coefficients. However, if the modifications required to the given blocks are too great, the block is simply regarded as invalid and not used for the embedding process. This is to ensure that the amount of distortion induced by the steganography algorithm is not noticeable by Wendy.

One of the most advanced algorithms using JPEG DCT coefficients is called the F5 algorithm. It relies on a random walk upon the quantized coefficients produced at the end of the quantization stage of JPEG compression. It was initially designed as an evolution to Jsteg, F3 and F4, but was later proved statistically breakable in [9].

The building block behind each of the algorithms mentioned above is still replacing the least-significant bit with a desired value. Thus, statistic attacks are possible in this circumstance. For example, Jsteg is falling short on the χ^2 statistical test, assuming a uniform distribution for the underlying message. This is to say that, given a histogram of JPEG coefficients, Jsteg *equalizes* the amount of even and odd coefficients.

The F5 algorithm was designed to specifically overcome the χ^2 attack. Nevertheless, it is still vulnerable the statistical attack proposed in [9]. The idea behind the steganalysis algorithm described therein lies in 2 steps:

- Estimating the original DCT coefficient histograms

- Estimating the probability of altering a DCT coefficient in the F5 process

The means of estimating the original DCT coefficients' histogram is performed by decompressing the JPEG compressed stego-object, cropping 4 columns of the resulting image, applying a low-pass filter (for instance a uniform blur), then compressing the resulting image using the same quantization tables as in the initial F5 embedding process and outputting the DCT coefficients' histograms. It was shown that the aforementioned method presents a high degree of similarity between histograms of the original cover image and the resulting histograms. Intuitively, since the DCT coefficients from the stego-object are computed on the image shifted by 4, they do not preserve the correlations they had before and may present quantitative artifacts of the original image and lose the spreading induced by the LSB embedding function. The low-pass filter applied in the second stage of the process is required due to the fact that a shift in the image may incur artifacts due to discontinuities of certain 8×8 blocks.

The conclusions reported in [9] show that the probability of false negatives in the process of detecting stego-objects with full-capacity embedding rate drops to 10^{-32} and to 10^{-7} in the case of the embedding rate being equal to 25%.

3.3 SSIS

The last approach to steganography discussed in this paper consists of spread-spectrum image steganography (SSIS). It was proposed by Marvel et.al. in [10] and is an example of robust steganography algorithm since it can withstand some amount of distortion before message loss. However, its attractiveness in this respect is drawn back by its high computational complexity. Since the results presented in the aforementioned article involve an important amount of techniques, the current paper will limit at presenting the intuitive aspects that revolve around the algorithm, its limitations and its technical challenges.

The main idea behind spread-spectrum lies in distributing information in several working frequencies of a given signal. Note that this technique is not novel, its usage in several communication technologies is well-established. The main idea behind spread-spectrum lies in broadening the frequency band of the input signal and the redistribution of information within the newly spread domain. Its main applications are due to the desire of resistance to jamming signals that might interfere with the input signal on a given frequency range.

The encoding algorithm takes as input the steganography key k, the input message P, the cover image C and outputs the stego-image S. Its main steps are depicted below:

```
key_1, key_2, key_3 = Schedule(k)
m=ErrorCorrection(Encrypt(key_1,P))
n=RNG(key_2)
gamma =Modulate(m, n)
```

```
I=Interleave(key_3, gamma)
S=Quantize(C + I)
```

Listing 1 Embedding procedure for SSIS algorithm

The scheduling function `Schedule` takes as input a character string and outputs 3 distinct keys. An example of this kind could be the key scheduler used for DES or AES or maybe the application of some hash function over some k-dependent bijections. The $Encrypt$ primitive symbolizes a symmetric encryption scheme, such as AES, DES et.al. `ErrorCorrection` is a primitive for creating an error-corrected output from an input bit stream. Examples of such codes include Hamming codes, Reed-Salomon codes et.al. The `Modulate` function serves to actually include the error-corrected encrypted signal in the random signal generated by RNG. It thus produces a low power signal that is fed into the `Interleave` block that spreads the incoming signal equally throughout the cover image such that the error-correcting code has equal chances of correcting words that are uniformly spread around the cover image. The concept of spectrum spreading is ensured by the modulation stage, whereby the broadband signal n is used as a carrier for m. If n is a Gaussian white noise with mean 0 and some variance, then a simple strategy is to simply encode each bit in m with -1 in case of a 0 bit or 1 in case of a 1 bit and then multiply the real number sequence outputted by the RNG with the respective positive or negative value. Note that the resulting signal also has a Gaussian noise distribution. However, the strategy presented above is inconvenient due to the low power of the Gaussian noise sequence, which results in small differences between the negative and positive values and may yield high errors at receiver-side. Therefore, a custom modulation scheme has been developed specifically for this ends, whereby a random uniform sequence is non-linearly modulated with a transformed version of itself, in such way as to maximize the minimum distance between an encoded 0 bit and an encoded 1 bit.

Upon receiver side, the reverse process is applied. Note, however, that several notions require further explanations. Let S' be the stego-image as seen at the recipient's end and k the shared stego-key.

```
(key_1, key_2, key_3)=Schedule(k)
I'=S′-Restore(S')
gamma'=Deinterleave(key_3, I')
n=RNG(key_2)
m'=Demodulate(gamma', n)
P=Decrypt(key_1,ErrorCorrectionDecode(m'))
```

Listing 2 SSIS Decoding

As per expectations, the steps depicted in Error! Reference source not found. are almost the reversed of the embedding algorithm Error! Reference source not found. but for a restoration step 2. The $Restore$ primitive is a procedure that estimates the original image prior to application of an additive Gaussian white noise. Examples of such algorithms include the Wiener filter or trimmed mean filters. The experiments conducted in [10] show that the best results in terms of BER (bit error rate) are obtained using the alpha-trimmed mean filter.

The results obtained using SSIS are remarkable in terms of resistance to known image processing techniques or noise addition. It was shown to resist JPEG compression of quality factor superior to 90% and would also resist Gaussian noise addition of power level up to 20. Nevertheless, one of the disadvantages of the proposed method lies in its high computational complexity, especially at recipient-end, mostly due to the complexity of the restoration filter.

In his paper from 2002 [11], Chadramouli shows a method of breaking the SSIS technique using an algebraic approach. In [12] another method for discriminating between SSIS modified images and regular ones is discussed while shifting the analysis method from the Kullback-Leibler distance proposed by Cachin [6] to Markov chains between adjacent pixels. In their analysis, they employ a SVM (state-vector machine) trained with a set of both regular and stego images in order to determine the features that change in terms of Markov matrices. The same approach is taken in [13], but this time relying on the correlations between different color components of the stego image. Their results show high detection rate with very small rate of false positives.

4 Test results

In the following section, several test results are presented. Note that even if multiple steganography tools are existing on the market – such as OpenStego [14], OpenPuff [15] etc. –, the approach taken in the current paper was to integrate in a common API (Application Programming Interface) the three steganography schemas depicted in the chapters above in order to employ a unified comparable testing framework.

The testing procedure consists of two parts: integrating the three implementations in a common framework and testing the three implementations for some common metrics. Python was chosen as programming language, due to its large support for image and signal processing in scientific environments.

The integration part consists in providing the following unified interface for a steganography schema:

```
Encode(Image cover, String message):Image

Decode(Image stego):String
```

Listing 3 Steganography Simplified API

The testing framework's purpose is to generate a set of results that reflect the amount of distortion induced by steganography to the cover images with respect to the length of the message embedded therein. As described in the previous sections, the SSIM (structural similarity), the MSE (mean squared error) and the PSNR (peak signal to noise ratio) metrics are employed to compare the result of

$Encode(cover, message)$ against the original $cover$and understand their dependence on $message$ length.

A secondary aim of the testing procedure depicted herein is to emphasize the trade-off one needs to take into account when performing SSIS in terms of distortion and the power of the superimposed noise comprised therein.

The results presented herein only apply to grayscale images, but similar tests can be envisaged for different color spaces.

For the aim of the current paper, the F5 implementation is that provided in [16].

The SSIS algorithm was implemented ground up, following the prescriptions specified in [10], with the exception that for the aim of current development, the restoration filter employed was the median filter with a kernel size of 3×3 .

In the following paragraphs, the results obtained during the analysis stage are presented.

4.1 SSIS analysis

The results obtained for the SSIS algorithm are highly dependent on the scaling factor (the Gaussian noise power) employed and on the length of the message. Note that the image analyzed is grayscale with pixel values ranging between 0 and 255. In **Fig. 16**, the bit-error rate is presented as a function of the superimposed Gaussian noise power. The payloads used for testing were randomly generated in order to simulate the modulation with a possibly encrypted payload.

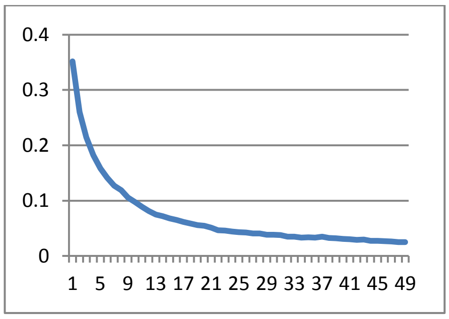

Fig. 16. Bit-error rate dependence on the power of the Gaussian noise

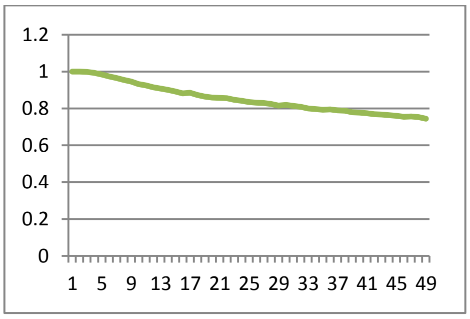

Fig. 17. SSIM dependence on the power of the Gaussian noise

In **Fig. 17**, the SSIM dependence on the power of the Gaussian noise is depicted. As can be observed, the SSIM drops linearly with the power of the superimposed signal, while the BER also decreases in this fashion. As expected, there is a trade-off between the amounts of distortion one induces while performing steganography and the retrieval possibility of the given schema. The same conclusion can be drawn from **Fig. 18**, where the PSNR dependency is depicted.

Note that the results obtained were given by the average values of the indicated metrics while varying the size of the message from 100 bytes to 1000 bytes.

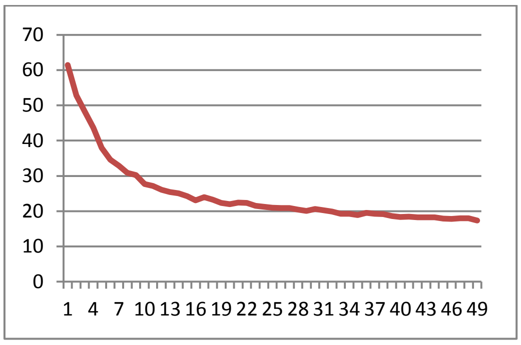

Fig. 18. PSNR dependence on the power of the Gaussian noise

4.2 Comparative analysis

In the following paragraphs the dependence of the distortion on the message length is analyzed. Thus, the curves corresponding to each algorithm are depicted and the results interpreted. For reference, the images that were taken into account when performing the mentioned analysis are part of the USC Viterbi dataset [17].

For simplicity considerations, even though the analysis was undertaken for all the aforementioned metrics: MSE, PSNR, SSIM, only the last two are depicted in the plots below. PSNR is used in favor of MSE because it entails the same amount of information, but it is normed against the dynamic range of the image to be analyzed and thus can provide a more qualitative view over the amount of distortion incurred.

In **Fig. 19**, the three curves represent the dependence of the PSNR on the length of the message. As can be seen, the best performing algorithm remains the LSB in the image space. However, judging by the derivative of the PSNR for the LSB and for the other two algorithms, one can derive the fact that for sufficiently big payloads, one might expect that the amount of distortion induced by LSB to become higher than for both F5 and SSIS. Note that for the SSIS, a scaling factor of 30 was employed for each image, as prescribed in [10] and as analyzed in 4.1. Hence, one can easily deduce that for lower scaling factors, with the incentive of the increase in BER, the PSNR is bound to decrease.

In **Fig. 20**, the same conclusions can be easily observed for the SSIM as for the PSNR. Hence, it can be deduced that LSB in the image space is the best performing algorithm with respect to the chosen metrics.

Note that even though the chosen metrics present a quantitative view on the amount of distortion induced to a particular image by a given algorithm with tuned parameters, as a function of payload length, they offer no guarantee on the security guarantees offered by the respective scheme. A more thorough approach is, hence, needed in order to emphasize the feature set that each algorithm modifies prior to drawing security assumptions.

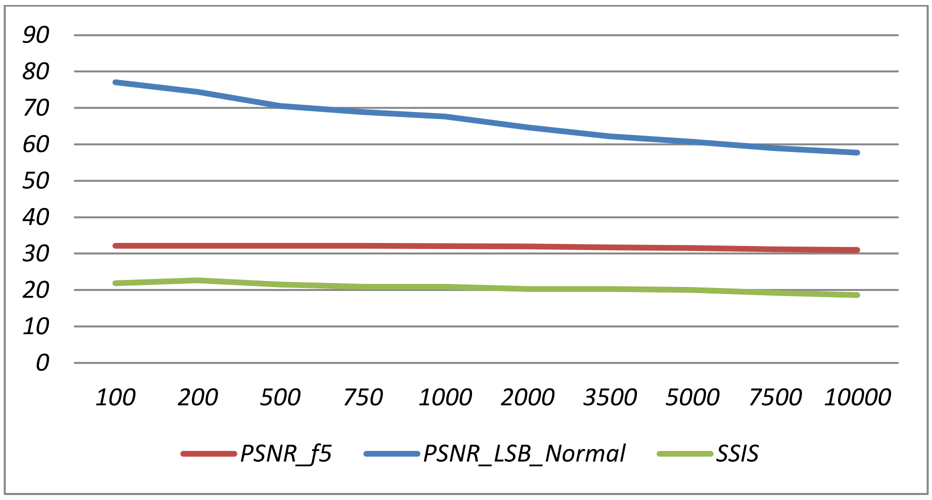

Fig. 19. PSNR vs Message Length

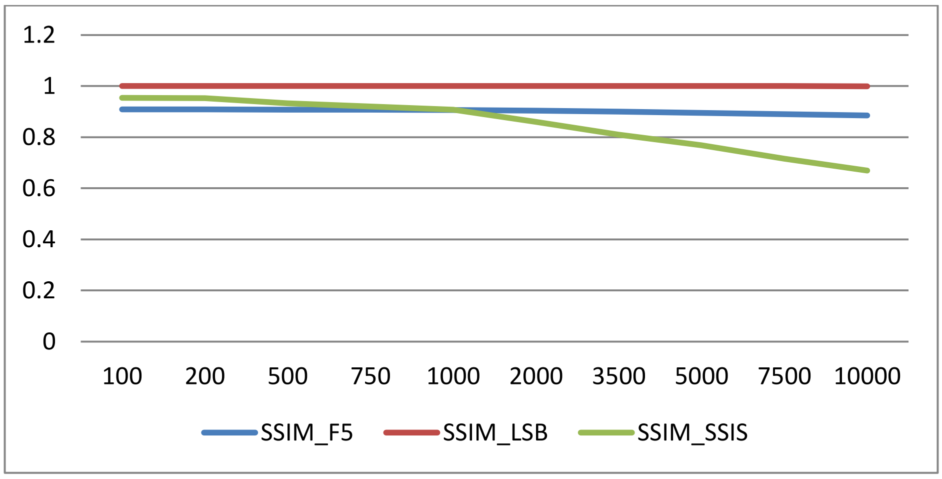

Fig. 20. SSIM vs Message Length

In **Fig. 21**, **Fig. 22**, **Fig. 23** and **Fig. 24**, the results obtained by applying different algorithms to a given image (namely $misc.camera()$ from the skimage package [18]), given message length of 10000 are presented.

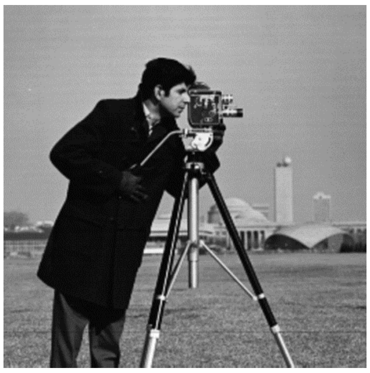

Fig. 21. Original image

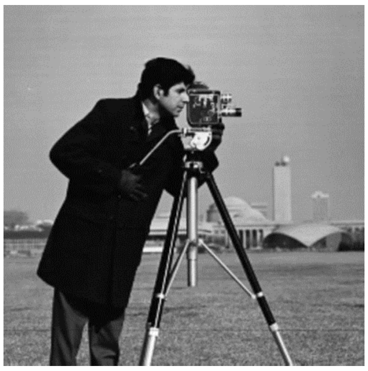

Fig. 22. LSB embedded message with length 10000

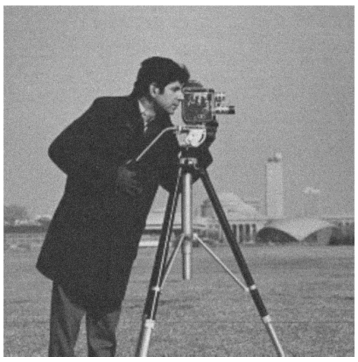

Fig. 23. SSIS embedded message of length 10000

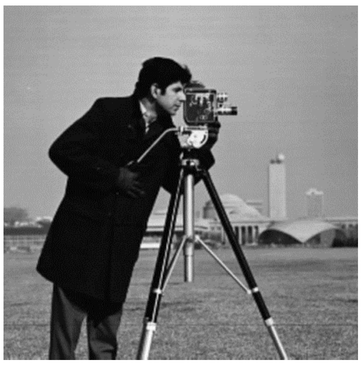

Fig. 24. F5 embedded message of length 10000

5 Conclusions and further work

As described, the purpose of the current paper was to perform an analysis over the most widely-used image steganography techniques, focusing on the different categories that comprise this discipline.

The insights into the topics presented in the current paper reside in a qualitative and, to some extent, quantitative assessment of three techniques employed in image-based steganography. As the current paper did not intend to be an exhaustive survey of steganography techniques, the three schemes described herein represent significant samples of the kinds of techniques employed in real life. The drawbacks and the attacks known for the current techniques were described in their corresponding sections.

As a general note, one may observe that all the steganography techniques described throughout the current paper are shown to be broken under some type of statistic or artificial intelligence-based tests. Several other techniques described in literature are known to be broken, while others have not yet undergone intensive steganalysis in order to be considered mature for industrial usage scenarios.

In conclusion, one can observe that steganography is a "young", highly multidisciplinary discipline – involving cryptography, image processing, signal processing, information theory et.al. – and undergoing active research in the last decades.

6 Acknowledgements

This work has been funded by University Politehnica of Bucharest, through the "Excellence Research Grants" Program, UPB–GEX. Identifier: UPB–EXCELENTA–2016 Research project title, Contract number 22/26.09.2017 (acronym: 406).

REFERENCES

[1] Herodotus și A. De Sellincourt, The Histories, Middlesex: Penguin Books Harmondsworth, 1954.

[2] I. J. Cox, Digital watermarking and steganography, Burlington, MA, USA: Morgan Kaufmann, 2008.

[3] S. Kaltzenbeisser și F. A. Petitcolas, Information Hiding for Steganography and Digital Watermarking, Norwood, MA, USA: Artech House, 2000.

[4] C. Shannon, „Communication Theory of Secrecy Systems," 1949.

[5] G. J. Simmons, „The prisoner's problem and the subliminal channel," în *Advances in cryptology: Crypto 83*, Santa Barbara, CA, USA, 1984.

[6] C. Cachin, „An Information-Theoretic Model for Steganography," *Information and Computation ,* vol. 192, nr. 1, pp. 41 - 56, 2004.

[7] A. Westfeld, F5—a steganographic algorithm., Berlin Heidelberg: Springer, 2001.

[8] J. Zhao și E. Koch, „Embedding Robust Labels Into Images For Copyright Protection," în *Proceedings of the International Congress on Intellectual Property Rights for Specialized Information, Knowledge and New Technologies*, Vienna, 1995.

[9] M. Goljan, J. L. Fridrich și D. Hogea, „Steganalysis of JPEG Images: Breaking the F5 Algorithm," în *IH '02 Revised Papers from the 5th International Workshop on Information Hiding*, London, 2002.

[10] L. Marvel, C. G. J. Boncelet și C. T. Retter, „Spread Spectrum Image Steganography," *IEEE TRANSACTIONS ON IMAGE PROCESSING,* vol. 8, nr.

8, pp. 1075-1083, 1999.

[11] R. Chandramouli, „Mathematical approach to steganalysis,” în *SPIE 4675, Security and Watermarking of Multimedia Contents*, 2002.

[12] K. Sullivan, U. Madhow, S. Chandrasekaran și B. S. Manjunath, „Steganalysis of spread spectrum data hiding exploiting cover memory,” în *IN SECURITY, STEGANOGRAPHY, AND WATERMARKING OF MULTIMEDIA CONTENTS VII*, 2005.

[13] J. J. Harmsen și W. Pearlman, „Steganalysis of Additive Noise Modelable Information Hiding,” în *Proceedings of SPIE - The International Society for Optical Engineering 5020*, 2003.

[14] OpenStego, „http://www.openstego.com/,” [Interactiv]. Available: http://www.openstego.com/.

[15] O. Steganography, „http://embeddedsw.net/OpenPuff_Steganography_Home.html,” [Interactiv]. Available: http://embeddedsw.net/OpenPuff_Steganography_Home.html.

[16] „jackfengji/f5-steganography,” 30 March 2012. [Interactiv].

[17] USC Viterbi, „SIPI Image Database,” USC Viterbi, [Interactiv]. Available: http://sipi.usc.edu/database/.

[18] „scikit-image: Image processing in Python — scikit-image,” [Interactiv]. Available: http://scikit-image.org/.