Steganalysis – Cybersecurity and National Defence

Alberto Eusebio, Francesco Imparato, Federico Morro,
Salvatore Orlando, Giulia Rivetti

Abstract

The aim of this paper is providing a general view of steganalysis declined in its aspects while giving insights on the steganographic tools used.

Steganography is the art of hiding information into a cover type (the mean of transmission used to hide the message) [1], whilst steganalysis is the art of analysing the cover type in order to detect the presence and in some cases even extract hidden information.

We will cover the state of art of the most important and used methods in steganalysis for different cover types: images (stego-key search, difference image histogram method, closest color pair method, JPEG steganalysis), audio (non compressed and compressed methods, phase coding, Mp3 steganalysis), text and TCP/IP.

Moreover, we will describe the general types of steganalysis that one can perform: statical, visual, spread spectrum, signature, transform domain, universal.

Finally, we will highlight the deep relation between steganography and steganalysis and describe their role of protagonists in the field of information security.

1 Introduction

Steganography is the technique of hiding a message within another message or a physical object.[1] The word *steganography* comes from the Greek word *steganographia*, which is the combination of *steganós* meaning "covered" and *-graphia* meaning "writing".

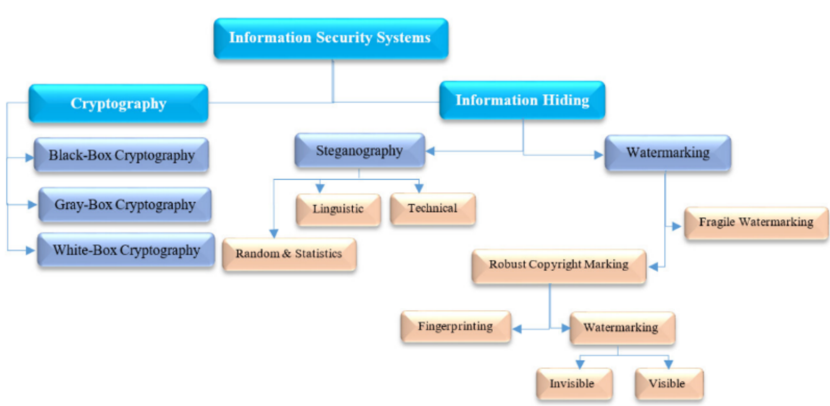
The first testimonials of the use of steganography date back to 440 BC in Greece mentioned by Herodotus in his *Histories* where he tells the story of Histitateus who tattooed a message on the shaved head of a servant of his and after the hair had regrown, he sent him to Aristagoras. Other historical exaples come from the story of Jeremiah Denton, who fell prisoner in 1966 during the korean war and encoded an help message in morse code, hidden in his blinking rhythm which transmitted during a TV report, and yet another example from microdots embedded in paper or in clothes used by espionage agents during and after the World War II.

With the advent of the *Digital Era* steganography has found fertile ground. Nowadays information can easily be concealed in almost every type of file in ways which are basically invisible to the common user. Images, videos and documents can contain hidden text as well as malicious code embedded into them, representing a serious threat for cybersecurity. Sometimes the hidden message could also be encrypted to be unreadable, even for active searchers. Although this pertains to the domain of cryptoanalysis.

In this paper we will focus on steganalysis, the study of detecting and, when possible, recovering hidden messages encoded using steganography. The ways in which it can be performed, examples of its application in known cases and the cybersecurity implications of its use in digital communication systems.

The Field on Information Security Systems is indeed vast and would deserve a much lengthier discussion. Here we attach a map which describes shortly the structure we will focus on.

Figure 1: Map of Information Security Systems [2]



2 Steganography

As we already stated, steganography is used to hide a message in a *cover type*. The latter can be any sort of information acting as a cover.

In all cases, the message is *embedded* into the cover type and becomes part of it. Differently from Watermarking where the mark is simply overlayed to the original message.

2.1 Methods

Steganographic methods may be grouped under two different classifications.

The first classification concentrates on the cover. We distinguish between injective and generative steganography. injecting steganography focuses on the methods implied in the injection of the *stego message* in the cover type. Generative steganography instead consists in the group of methods aimed at constructing a cover from the stego message.

The second classification is oriented on the mathematical representation of the cover types(treated as signals). Heere we distinguish between substitutive and constructive steganography. Substitutive steganography targets the background noise in the signals. In this case the noise is modified to embed the message. This usually can be achieved by exploiting the *LSB*[1] of each byte, which is most susceptible to error, to hide the message in the signal matrix.

Several drawbacks arise, such as the limited size of the message to be inserted to avoid amplified distorsions of the original signals, the resilience to various degrees of compression and signals transformations, and lastly the transparency of the message. We leave for a further discussion the different available tradeoffs, selectively mitigating some of the aforementioned drawbacks.

Similarly, constructive steganography integrates a noise model into the message. The main drawback is that it is difficult to produce, and by design fragile to attacks[2].

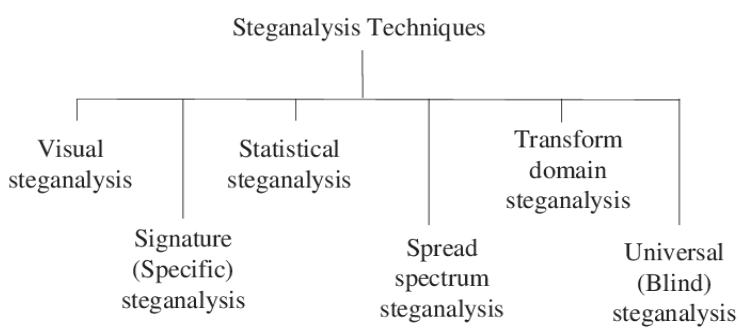
3 Steganalysis

Steganalysis is the branch of study dedicated to analysing the methods and the vectors used to transmit a hidden message in order to retrieve such message whenever it is present. Usually we call steganalysis successfull when it is capable of detecting the presence of a *stego message*. Unlike cryptanalysis where the message may even be apparent but encrypted, in steganalysis the study of the message starts from a suspect.

3.1 Methods

Hereinafter we will present the methods used in steganalysis in order to obtain the secret message. The following image illustrates all of them. Here we simply want to introduce them in a rigorous way, but they will be treated in the specific cases when dealing with the cover types.

Figure 2: Steganalysis techniques



3.1.1 Statistical

Statistical steganalysis consists in using tools borrowed by Statistics in order to spot anaomalies in the cover message. These anomalies can be usually detected by searching in the data for patterns or by simply analysing some alteration of the modelled probability distribution of the data in the message. Often Neural Networks are involved in this type of steganalysis.

[1]Least Significant Bit

[2]When referring to attacks to steganography we intend the steganalysis techniques involved to detect the message

3.1.2 Visual

This is a very rudimental yet effective way of determining if a message was modified or not. This technique relies on the human eye and its ability to spot anomalies in the format of a message which may raise suspicions.

3.1.3 Spread Spectrum

This is a technique borrowed by signal analysis which consists in treating the data on which we want to perform the analysis as a signal. Following this reasoning it is almost immediate to see that the stego message embedded into our data is nothing but noise in our signal. Finally the steganalysis is performed in this case just like any other type of signal analysis.

3.1.4 Signature

Signature is deeply correlated to the field of Watermarking. In many cases the data on which it was performed the steganography could have been signed. The insertion of the stego message inside the signed data may thus lead to the alteration in the digital dignature, making the message visible to the receiver.

3.1.5 Transform Domain

Even in this case the goal is to start from considering the data we are analysing as a signal. Then we try to transport this signal from a domain into another so that to make it invisible to the end user.

3.1.6 Universal

This is a type of analysis performed in a sort of blind mode. In this case we don't care about the medium through which the message was sent but we perform a series of operations(reading LSB for example) which has little to nothing to do with the message itself.

3.2 Cover types

In the following sections, we will treat several *cover vectors* to which steganography can be applied in order to hide data. In particular we will focus on some of the most common methods the steganalytic techniques involved in detecting and in some cases even in retrieving the embedded information. Notice that since different cover types require different approaches due to their intrinsic characteristics, we have decided to dedicate a chapter to each of them.

4 Images

Images are likely to be chosen to hide secret messages due to the low sensibility of the *human visual system* (HVS) to some particular attributes such as small changes in luminance or brightness or contrast near figures edges. There are several methods to apply steganography to an image, in the following sections we will treat some of the most common hiding techniques and some of the steganalysis methods to deal with them.

4.1 Stego-key search and encryption

It is important to notice that when we refer to modifications of the *LSB* in images we do not necessarily imply modifications to every byte. Whenever allowed by the size of the cover type, it is more useful and dicrete to follow some pseudo-random sequence. This sequence is usually decided by a *stego-key* (usually stego-key will be mapped in a set of possible seeds by a hash map) [3].

We also want to point out that the embedding of information could also be preceded by an encryption of such information. This procedure makes the extraction of the message through a brute fore approce practically unfeasible: the complexity would be proportional to the cardinality of the set of possible seeds time the one of the set of possible encryptions. Although the encrypted message is still susceptible to attacks.

4.2 LSB embedding method

LSB embedding method is arguably the most popular steganographic method, due to its simplicity, high imperceptibility and high capacity. In this method, the image is decomposed into *bits plane* (8 bits per pixel for grayscale and 24 for color images, one for each color channel) and the *least significant bit* (LSB) is substituted with the message to be hidden. Note that even if the message is encrypted, due to its simplicity, this method is easily detectable with a statistical steganalysis attack [4].

4.3 Difference Image Histogram method

The *Difference Image Histogram method* is derived by the easier idea of analysing the *histogram distribution* of a natural image and its stegoimage. When we are steganalysing an image we usually do not have the cover image. What we can do is comparing the histogram distribution of the stegoimage with a set of images similar to the one. By the law of large numbers we should be able to find a *mean imgae* to use to compare with our stegoimage. The criticalities arise when the set is polluted. [5]

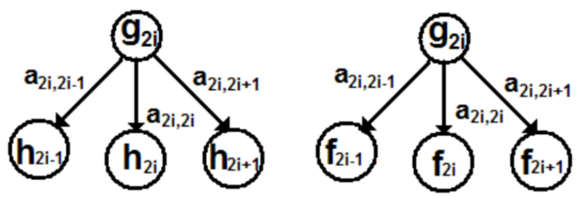
The proposed way to proceed is the following [6]. Starting from the test image (that we will call h, considering h a grayscale image or a color image under some assumptions) we initially generate an image f given by h with flipped LSB. Then an image g given by h zeroing the LSB. Finally images D_h, D_f, D_g, created by the respective image of each one with the following formula:

$$D_I(i,j) = I(i+1,j) - I(i,j)$$

where I is the image denoted by the subscript and the couple and (i,j) a unique pixel of the image. At this point, we can analyse the histograms of the three generated images D_h, D_f, D_g:

$$H = \{h_i | i = -255,\ ...,\ +255\},\ F = \{f_i | i = -255,\ ...,\ +255\},\ G = \{g_i | i = -255,\ ...,\ +255\}$$

Figure 3: Transition values from G to H and F



then, we define the following values: $\alpha_i = \frac{a_{2i+2,2i+1}}{a_{2i,2i+1}}$ $\beta_i = \frac{a_{2i+2,2i+3}}{a_{2i,2i-1}}$ $\gamma_i = \frac{g_{2i}}{g_{2i+2}}$

As described by X. Ping and T. Zhang in [6]:

- if $\alpha_i \approx 1, \forall i \in \{-255, -254, ..., 255\}$ then the image contains some hidden message;
- otherwise, for natural images, $\alpha_i \approx \gamma_i$ is satisfied.

4.4 Closest Color Pair method

Another method used to detect hidden messages on the LSB plane is the *Closest Color Pair method.*

When an image has a steganographed message in te LSB plane, the number of close colors increases [7]. Given two pair of colors $C_1 = [R_1, G_1, B_1],\ C_2 = [R_2, G_2, B_2]$, the condition of them being close is:

$$(R_1 - R_2)^2 + (G_1 - G_2)^2 + (B_1 - B_2)^2 \leq 3$$

We apply a LSB embedding steganography algorithm on the image and we compute the number of close color pairs in both images.

Now, we define:

- P as the number of close color pairs; P' as the number of close color pairs of the stegoimage;
- U as the number of color pairs; U' as the number of color pairs of the stegoimage

Then, we compute:

$$R = \frac{P}{\binom{U}{2}},\ R' = \frac{P'}{\binom{U'}{2}}$$

We know that if

$$\frac{R}{R'} \geq Th,\ Th = 1.1$$

then the image is a natural image, otherwise it contains an hidden message. All the proofs are in [7].

4.5 JPEG steganalysis

JPEG images are one of the most used format on Internet web sites, due to their high compression rate, while maintaining a good quality.

4.5.1 DCT Domain embedding methods

The DCT Domain embedding methods modify the compression coefficients in order to hide data inside the image. The JPEG format uses a *discrete cosine transform* (DCT) to transform every 8x8-pixel block into 64 DCT coefficients, that are used to calculate the pixels when the image is displayed. The simplest and most used DCT Domain embedding method substitute the *LSB* of the coefficients with the secret message: since the modification is done in the frequency domain, there is no human perceivable change in the image. However, these modifications can be detected by analyzing the DCT coefficients which changes significantly with respect to a natural image [8].

4.5.2 Chi-square test

The Chi-square test is a statistical steganalysis test, which aims at determining whether an image shows distorsion from embedding hidden data.
Let n_i be the frequency of the DCT coefficient i in the image, we assume that an image with hidden data has similar frequency for adjacent coefficients so we compute the arithmetic mean $y_i^* = \frac{n_{2i}+n_{2i+1}}{2}$ to derive the expected distribution. The expected distribution is compared with the observed one $y_i = n_{2i}$.
The chi-square distribution for the difference between the expected and the observed DCT coefficients is calculated as follows:

$$\chi^2 = \sum_{i=1}^{\nu+1} \frac{(y_i - y_i^*)}{y_i^*}$$

where ν are the degrees of freedom, which are one less than the categories in the DCT coefficients histogram.
The probability that there is an embedded message can be computed as the complement of the *cumulative distribution function* of the chi-square distribution.

Note that, as presented in the stego-key section, different algorithms modify the coefficients not sequentially or following different orders, so the steganalysis process is usually performed by calculating the probability of the presence of an embedded message considering different portions of the image at the same time [8].

5 Audio

Steganography in audio is more challenging with respect to images, because the *human auditory system* (HAS) operates over a wide dynamic range, while maintaining a high sensitivity to perturbations and noises. Howerer, there are still some "holes" where data can be hidden. The HAS has a quite small differential range, where loud sounds mask out quiet sounds, and is unable to perceive absolute sound phase, but only relative one.

Moreover, another important factor to consider when dealing with sound are the transmissions environments. Audio signals can be transmitted through a digital channel (eventually being resampled), through an analog channel or "over the air" played by a speaker and received by a microphone. Depending on the transmission channel there could be huge modifications that can make the steganalysis process impossible, but also that can compromise irreparably the hidden message, damaging also the steganographer [4].

In the following sections we will cover some of the possible ways in which steganography and steganalysis is applied in audio signals.

5.1 Non-compressed and compressed methods

We can distinguish two different types of audio steganography: steganography on *non-compressed* audio files and on *compressed* ones. The first ones aims at exploiting the vulnerabilities of the *HAS* presented above, whereas the second ones perform minor modifications to embed data based on the way in which the compression is performed [9].

5.2 Phase coding

The phase encoding method works by modifying the *absolute phase* of audio signals to convey information, while maintaining the *relative phase* in order to not compromise imperceptibility.

5.2.1 Encoding procedure

Firstly the sound is sampled into N short segments. To each one of them a *discrete Fourier transform* (DFT) is applied, constructing two matrices: $\phi_n(\omega)$ for the phase and $A_n(\omega)$ for the magnitude. Sequently the phase difference between ajacent segments is stored and the message to hide is converted into a phase message, representing with $\frac{\pi}{2}$ or $-\frac{\pi}{2}$ respectively 0 or 1. The phase matrix is then recomputed by embedding the message into it. Finally the original message is reconstructed applying an inverse DFT.

5.2.2 Steganalysis techniques

The steganalysis techniques for phase coding system follows a procedure similar to the encoding ones, since is based on a statistical analysis of *phase discontinuites*, but which requires the use of a classification algorithm in order to distinguish between natural and modified signals.

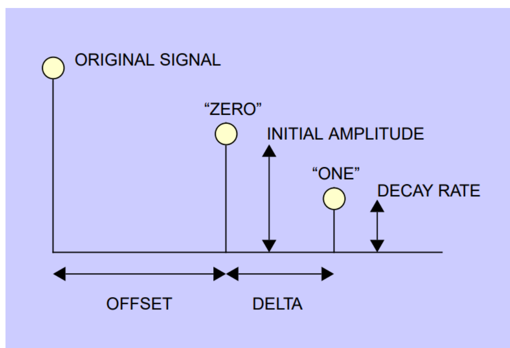
As proposed in [10], the signal is divided into segments to which is applied a *fast Fourier transform* (FFT) in order to extract the phase difference between adjacent segments. Such phase difference is then monitored, calculating statistical features of the sample. Finally a *SVM classifier* (support-vector machines) is used to detect if the signal has been modified or not.

A SVM classifier, after a proper training, is able to clearly distinguish between elements by mapping them into two different classes by means of linear regression and statistical calculus. This method is applied varying the length of segements to which the FFT is applied.

5.3 Echo embedding methods

The echo embedding methhods hide data introduzing or varying an *eco*. Data are hidden by varying *initial amplitude*, *decay*

Figure 4: Echo kernels and parameters



rate and *offset*.

5.3.1 Encoding

Encoding procedure is performed by dividing the signal into smaller portions and the echoing each portion as an independent signals by an audio mixing process. The coder use two different delay times to represent a binary zero(offset) and a binary one(offset+delta), both below audible threshold, considering the initial amplitude and that the decay rate follows and exponential behaviour. The final signal is formed by recombining all independent encoded signal portions. The transitions between one and zero are done by slightly modify the zero and one kernel in order to maximize imperceptibility.

5.3.2 Steganalysis

The steganalysis process of echo encoded signals is done by doing a statistical analysis on the *cepstrum*, which is the result of the Fourier transform on the decibel spectrum of the signal. As presented in [9], the cepstrum is calculated in a sample window possibly smaller than the encoding segment length. The sampling window is moved over the length of the signal and the cepstrum recomputed each time. Then, the results can be analyzed by classifying every sampled cepstrum in one of four possible categories:

- Inside a zero embedded segment
- Inside a one embedded segment
- Crossing from a one(zero) to a zero(one)
- Crossing from a one(zero) to a one(zero)

This classification is possible due to the fact that the cepstrum plotting exhibits peaks when encountering a delay defined by the 0 or 1 kernels. Moreover, cepstrum peak location aggregation rate (CPLAR) is introduced as the ratio between detected peaks and number of sampled windows. CPLAR is used to discriminate between natural and steganographed audio signals. This method is also capable of detecting the length of the segmentation used by the coder.

5.4 Mp3 steganalysis

The last type of audio steganalysis that we will briefly treat regards the *Mp3* format, which is one of the most used compressed sound formats, since it provides a high compression rate and a good quality.

The Mp3 compression algorithm consist of two nested loops. The *inner loop* does the quantization of the data and determines the suitable quantizer step in function of the available quantity of bits. Whereas the *outer loop* controls the distorsion of the encoding and keeps it beyond the perception level.

The most common steganographic algorithms apply some modifications on the encoding algorithm, for instance, modify the termination condition on the inner loop and hide the data during the compression process or modify the compression coefficients or parameters saved (for instance by replacing the LSB).

To perform a steganalystic process on an Mp3 file it is necessary to do a statistical analysis on the length of the quantization steps or on the *MDCT* (modified discrete cosine transform) coefficients, the transform used in the compression algorithm. The analysis methods are similar to ones presented in the other sections in this paper, since they rely on some previous knowledge of what is expected and on calculations that will tell if there is or not an encrypted message.

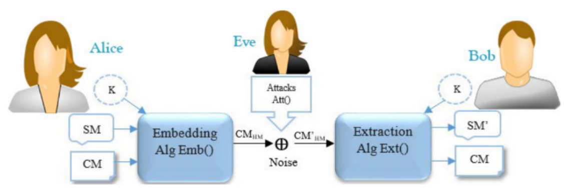
6 Text

In this section we will illustrate in general the principle of work of the MTHS using the famous Simmons' prisoner problem. The information conveyed in this part refers mainly to [2].

6.1 Modern Text Hiding Schema

The scenario proposed by Simmons present two prisoners, Alice and Bob who wishes to communicate, and Eve an active warden who tries to disturb the communication.

Figure 5: Modern Text Hiding Schema



The proposed image intuitively explains the logic of the schema. Alice who whishes to send a Secret Message(SM) to Bob, will produce a Cover Message(CM), *i.e.* an innocent message which will work as a carrier. She will then embedd the SM into the CM using a steganographic algorithm and, optionnally[3], securing it with a key(K). In the end she will pass the message to Eve, who will analyse it before giving it to Bob. Eve, an active warden, will use steganalysis tools in order to break the cover taylored by Alice and will possibly distort the message in order to make it unreadable to Bob. Once the message arrives to Bob, he will apply the inverse steganographic algorithm in order to read the message. Since the message could be corrupted Bob will have to perform some error correction in order to make the message readable again(whenever this is possible). To summarize the principal characteristics of the message sent by Alice are:

- *invisibility*: the message should not be noted by Eve
- *capacity*: the CM must be large enough to embed all the SM
- *distortion robustness*: the SM should resist the noise produced by Eve in the channel

6.2 Steganography in text

Now let us abandon the prison world to move to the digital domain. We now want to propose intuitively some algorithm which could be exploited to cover hidden messages into text. The field of hidden-text encoding owes its growth to two main factors: the introduction of the Unicode Standard and the growth in popularity of social media and messenger applications. The number of "invisible" characters introduced by the Unicode standardization can be exploited by steganographic algorithms just as invisible ink was exploited in handwritten steganography. Combining this result with the fact that the modern society produces every year more data than ever before in human history, the possibilities and the threats brought by steganography in the digital world are basically limitless.

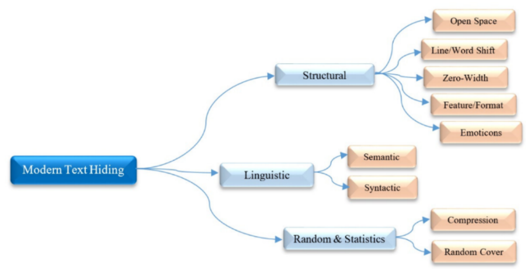
As you can see the field is vast even here and unfortunately we will only treat the most interesting points of the numerous ramifications.

6.2.1 Structural

This set of techniques consist in altering the structure of the document rather than actively modifying the file. The case of *open space* is significant: the spaces are substituted by other invisible Unicode characters. This technique is easily

[3]This is the domain of cryptography

Figure 6: Text Hiding techniques



noticeable(even by the human eye) and not so much robust. Similar is the *zero width* techniques which exploits the zero width characters present in the Unicode standard. Even more visible are the format, color and alignement alterations of *Format*. Much more interesting is the case of *Emoticons* which has found secure ground in the last period and which consist of assigning to each emoji a code(letter or word).

6.2.2 Linguistic

Much more stable are the algorithms which exploits linguistic characteristics in order to hide the presence of hidden bits. In the *Semantic* case the message is hidden inside abbreviations or achronyms. This method provides much more reliability and lower embedding capability than the previous one. Much worse and visible is the *Syntactic* approach which consists in changing symbols with others to represent some code.

6.2.3 Random and Statistic

These method start only from the SM and try to generate a CM. They are usually much more time and power consuming. The *Compression* method consists in compressing the text of the SM using well known compression algorithms such as *Huffman Coding*. The compressed text is then reordered to mimic the user-typed format or at least a plausible one. Not much more efficient is the *Random Cover* approach where an algorithm(*AH4s*) is used to generate a long(and usually meaninglsess) text from the specified SM. The latter method is indeed very visible and also computationally expensive.

6.3 Steganalysis

6.3.1 Visual steganalysis

This is a human powered technique which simply consists in reading the texts searching for anomalies in the syntax, layout and type of characters used. This type of analysis can only be performed by trained users and in most of the scenarios simply detects the presence of hidden messagges without retrieving it. A common type of attack which could be performed by the user consciously(not necessarily) is manually changing the structure of the message. This may lead to the loss or irretrievability of the SM.

6.3.2 Statistical steganalysis

We have already treated this in the previous section, but now we will focus our attention on the particular case of text steganography. As the aim of this paper is not to dwell excessively in any mathematical rigour, we will only attempt to give an intuition of the underlying mathematical framework that supports it. The basic principle beneath is the assumption that it is possible to describe the message with a probability function. Recalling some basic principles of probability, if $x_1, x_2, x_3 \dots x_n$ are independent, then $p(x_1, x_2, x_3, ...x_n) = p(x_1)p(x_2) \dots p(x_n)$. Instead when they are not then

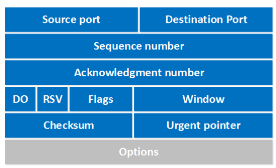
$$p(x_1, x_2, x_3, ...x_n) = p(x_1)p(x_2|x_1)p(x_3|x_1, x_2) \dots p(x_n|x_1, x_2, \dots, x_{n-1})$$

Starting by the assumption that written text follows a logic schema and thus that words are interdependently related, then by computing the probability function of the text and comparing it to the one of the *stego-message*(assumed to be perturbed by steganography) we can detect the presence of steganography in the text. This tells us that in theory it is possible to detect whether a sequence of characters follows a specific pattern or not, and it is in theory also possible to retrieve the message from such prediction. In [11] the experiment presented shows that it is possible to use a neural network to recognize patterns into text messages which are proper of *stego-messages*.

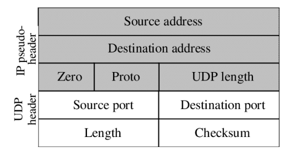
7 TCP/IP

7.1 Introduction

Before treating such topic we must briefly discuss the behaviour of the TCP/IP (**Internet protocol suit**) communications. The *IP (Internet Protocol)* is at the basis of internet communication. The prnotocol exploits the principle of encapsulation, sending packets composed by a *header* which contains information such as the **destination address** and **source address** and a *payload* which represents the actual data to be transmitted. Since physical channels usually have a limited *MTU*[4], then the data is usually fragmented in smaller chunks, wrapped into an header and only then transmitted. The *TCP (Transmission Control protocol)* is derived protocol(from IP) used for reliable (little information loss), ordered, error-checked data transmission between a machine hosting the data (**Server**) and another requiring such data (**Client**). It is performed previous a three-way handshake which estabilishes a connection between server and client, thus preparing a reliable communication channel. Contrary to TCP is *UDP (User Datagram Protocol)*, which is far less reliable but faster. It broadcasts the data in an unordered and uncontrolled way to the receiver. There is no need for estabilishing a connection in order to implement such protocol.



(a) Structure of TCP packet



(b) Structure of UDP packet

7.2 Exploits and protection tools

The first part of the header is part of the IP protocol. In the second part we find different types of metadata which are specific for the protocol. Generally the latter is redundant and such redundancy could be exploited by a *stego-algorithm*[5] which starting from a *cover-network packet sequence*[6] and a *covert message*, can generate a sequence of packets(each one embedding a portion of the covered message) which will be sent over the network.

Despite not being *risk-free* since the message could be corrupted in the meanwhile, or even detected while traversing the multiple nodes of the net, such procedure is widely used.

Hereinafter we illustrate a series of procedures acted with the purpose of defending from this type of attacks.

7.2.1 Tos (Type of service)

Tos are a field in the TCP header which are nowdays rarely used. This could open doors to a steganographic attack if modern operating systems did not set them to zero by default. A warden monitoring the channel could immediately signal an error.

7.2.2 IP ID

IP ID is a field in the Internet Protocol which is used to assist the receiver in reassemblying the fragmented packets. This field consist of randomly unique numbers representing a packet. It is possible to insert other types of information in this field by simply conform to the uniqueness constraint. Since in many cases the numbers used for the IP ID are not random, by knowing the charachteristics of the sender it is possible to detect an infiltration. Many networks use a *Sequential Global IP ID* for network communication which uses a global counter for the IP ID. An active warden could simply check that $IP_{id2} = IP_{id1+1}$ to verify the cosistency of the message. Other systems use instead a *Sequential Per-host IP ID* which is nothing but a per host seqeantial counter. Therefore the defence protocol is basically the same. A slightly different technique consists in the *IP ID MSB Toggle* in which the *OS* toggles the most significant bit of the IP ID every *rekey interval*[7], so that the warden can examine the MSB to check if it matches this pattern. Since the IP ID is unique, then it is common to check for non repetance of IDs.

7.2.3 IP Fragment Offset

IP Fragment Offeset is an offset which is present in the IP header which helps the receiver to reconstruct the sequence of bits from the fragmented sequence. Modulating the size of the fragments changes the offset field in the header and thus a

[4]Maximum Transmission Unit
[5]an algorithm used for steganography
[6]a sequence of packets which will be transmitted over the network acting as a cover for the hidden messace
[7]The WPA protocol generates periodically a security key

message could be sent. The protection against this method is simply checking the size of the packets relative to the MTU and so even in this case a warden can easily detect an error.

7.2.4 TCP sequence number

The TCP sequence number is a field in the TCP header which stores the randomly chosen position(for security reasons) of the first byte to be transmitted through the channel. The steganographic method consists in replacing this field with the data to be sent. Being random it is more difficult to spot a breach in the channel. In this case the usage of a *SVM*[8], a machine learning tool able to identify patterns inside the data transmitted could come into hand. However an error could be detected even simply by checking the presence of repetitions in the stream(not admitted by design).

7.2.5 Timestamp modulation

Timestamp modulation is another technique of steganography which operate by modifing the *LSB* of the timestamp of a TCP packet in order to represent a '1'. The covert message is thus embedded into the data stream. Since the TCP Timestamp support is not universal, machines not supporting such feature may detect the hidden message. Since there could be errors in the transimmssion of these packets, there is a threshold of randomness which must be taken into account and which could lead to some unexpected results.

7.3 Other techniques

There are also other methods which are simply listed(and not explained) which search for errors by checking *unsual flags*, *excessive fragmentation*, use of *IP options*, *unexpected TCP options* and *excessive re-ordering*.

8 Conclusions

Up to now we discussed what steganography and steganalysis are and how the latter is performed. The reason why steganalysis is important is that nowadays almost everyone uses a computer or a digital device, and the security of informations is a primary object of attention. This attention led to a development of studies on signal processing jointly with information security services. Steganography is not the only way one can hide informations, there are multiple options, each one of them with a specific target:

- encryption: confidentiality
- watermarking: copyright protection
- steganography: privacy that can also prevent traffic analysis

We can also make an example to clarify the difference between encryption and steganography: *the prisoners' problem* [12], which considers that two accomplices in a crime have been arrested and are in two different rooms. The warden decides to let them communicate by exchanging written messages at the condition of reading all of them, hoping to find some information and because he fears that they could share an escape plan. Now, if the two prisoners encrypted their messages the warden would have been able to notice that they are communicating something "strange" and probably he will not deliver the message to the other criminal; in the case in which the two prisoners steganographed the message, the warden would not have been able to find that there were hidden information.

Given this example, we can also clarify what *steganalysis* is: it can be seen as the process that the warden should do in order to find that in the "normal" message of the prisoners there were hidden information.

At this point we can easily understand that the steganography method is "*broken*" simply when steganalysis is able to detect that in the cover type there is a hidden message. In general we can also distinguish between

- *passive steganalysis* which simply detects the message without knowing anything else
- *active steganalysis* which detects the message with some extra information such as the length of the message and/or its location

Moreover, when steganalysis is performed successfully, we can also update the steganography method which has been broken by the steganalysis. This continuous research in steganography and steganalysis is useful when it comes to protect data. [9]

[8]Support Vector Machine

References

[1] Merriam Webster. *Steganography definition*. URL: https://www.merriam-webster.com/dictionary/steganography.

[2] Milad Taleby Ahvanooey, Qianmu Li, Jun Hou, Ahmed Raza Rajput, Chen Yini. "Modern Text Hiding, Text Steganalysis, and Applications: A Comparative Analysis". In: (2019).

[3] D. Soukal J. Fridrich M. Goljan. "Searching for the Stego-Key". In: (2004).

[4] N. Morimoto A. Lu W. Bender D. Gruhl. "Techniques for data hiding". In: (1996).

[5] A. Hernandez-Chamorro, A. Espejel-Trujillo, J. Lopez-Hernandez, M. Nakano-Miyatake, H. Perez-Meana. "A Methodology of Steganalysis for Images". In: (2009).

[6] X. Ping T. Zhang. "A new approach to reliable detection of LSB steganography in natural images". In: (2003). DOI: https://doi.org/10.1016/S0165-1684(03)00169-5.

[7] R. Du J. Fridrich M. Goljan. "Detecting LSB Steganography in Color and Gray Scale Image". In: *IEEE Multimedia* 8.4 (2001).

[8] P. Honeyman N. Provos. "Detecting Steganographic Content on the Internet". In: (2001).

[9] M. H. Kayvanrad H. Ghasemzadeh. "Comprehensive Review of Audio Steganalysis Methods". In: (2017).

[10] R. Hu W. Zeng H. Ai. "A Novel Steganalysis Algorithm of Phase Coding in Audio Signal". In: *IEEE* (2007).

[11] Zhongliang Yang , Yongfeng Huang , and Yu-Jin Zhang. "A Fast and Efficient Text Steganalysis Method". In: (2019).

[12] G.J. Simmons. *The Prisoners' Problem and the Subliminal Channel*. Springer, 1984. DOI: https://doi.org/10.1007/978-1-4684-4730-9_5.