<http://es.wikipedia.org/wiki/Esteganograf%C3%ADa>

La **esteganografía** es la disciplina en la que se estudian y aplican técnicas que permiten el ocultamiento de mensajes u objetos, dentro de otros, llamados portadores, de modo que no se perciba su existencia.

Si bien la esteganografía suele confundirse con la [criptografía](http://es.wikipedia.org/wiki/Criptograf%C3%ADa), por ser ambas parte de los procesos de protección de la información, son disciplinas distintas, tanto en su forma de implementar como en su objetivo mismo. Mientras que la [criptografía](http://es.wikipedia.org/wiki/Criptograf%C3%ADa) se utiliza para cifrar o codificar información de manera que sea ininteligible para un probable intruso, a pesar del conocimiento de su existencia, la esteganografía oculta la información en un portador de modo que no sea advertido el hecho mismo de su existencia y envío. De esta última forma, un probable intruso ni siquiera sabrá que se está transmitiendo información sensible.

Sin embargo, la [criptografía](http://es.wikipedia.org/wiki/Criptograf%C3%ADa) y la esteganografía pueden complementarse, dando un nivel de seguridad extra a la información, es decir, es muy común (aunque no imprescindible) que el mensaje a esteganografiar sea previamente cifrado, de tal modo que a un eventual intruso no sólo le costará advertir la presencia de la mensajería oculta, y si la llegara a obtener, la encontraría cifrada.

la esteganografía ha estado presente desde tiempos antiguos y constantemente va evolucionando.

Se usa en documentos, imágenes, audio, vídeo, etc.

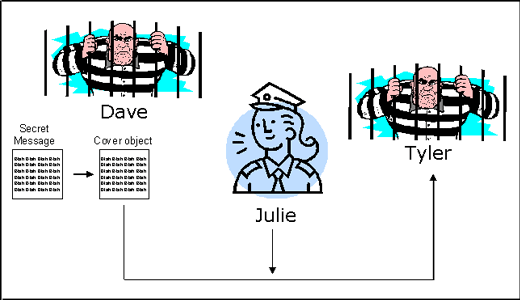
Lo que la esteganografía esencialmente hace es explotar las limitaciones de la percepción humana

Debido a que la esteganografía es invasiva, es decir, deja huellas en el medio de transporte utilizado, las técnicas de esteganálisis se basan en cómo detectar estos cambios.

http://www.snotmonkey.com/work/school/405/#

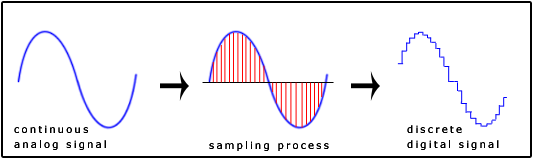
In today's digital age, steganography has surfaced once again as a relatively new technology used in computer security.

Groups ranging from those interested in protecting copyrights to people motivated to protect privacy have shown great interest in the applications of steganography.



AUDIO DIGITAL

Digital audio differs from traditional analog sound in that it is a discrete rather than continuous signal. A discrete signal is created by sampling a continuous analog signal at a specified rate.



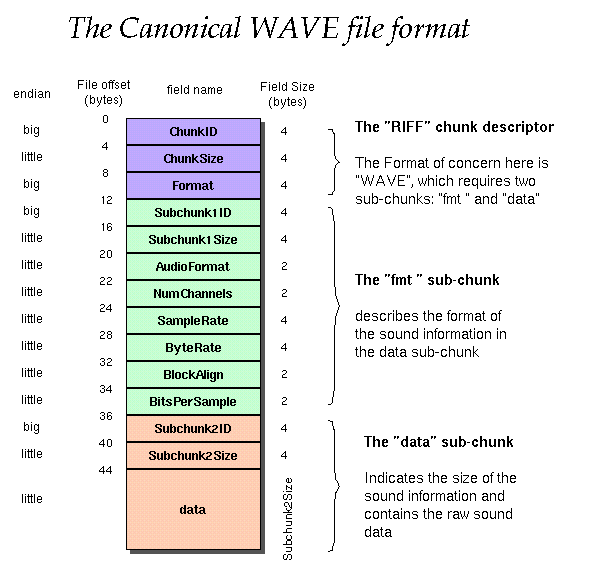
Digital audio is stored on a computer as a sequence of 0's and 1's. With the right tools, it is possible to change the individual bits that make up a digital audio file. Such precise control allows changes to be made to the binary sequence that are not discernible to the human ear.

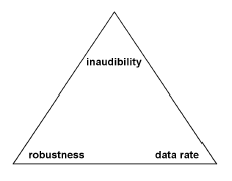
FORMATO WAV

<http://es.wikipedia.org/wiki/Waveform_Audio_Format>

**WAV** (o **WAVE**), apócope de *WAVEform audio file format*, es un formato de audio [digital](http://es.wikipedia.org/wiki/Se%C3%B1al_digital) normalmente sin [compresión de datos](http://es.wikipedia.org/wiki/Compresi%C3%B3n_de_datos) desarrollado y propiedad de [Microsoft](http://es.wikipedia.org/wiki/Microsoft) y de [IBM](http://es.wikipedia.org/wiki/IBM) que se utiliza para almacenar sonidos en el [PC](http://es.wikipedia.org/wiki/Computadora_personal), admite archivos [mono](http://es.wikipedia.org/wiki/Mono) y [estéreo](http://es.wikipedia.org/wiki/Est%C3%A9reo) a diversas resoluciones y velocidades de muestreo, su extensión es .wav.

<https://ccrma.stanford.edu/courses/422/projects/WaveFormat/>



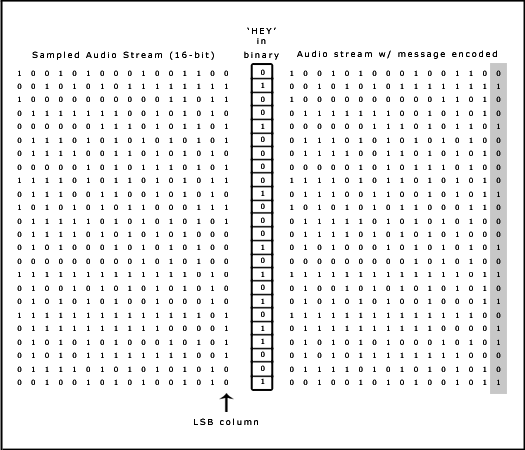


## (A PARTIR DE AQUÍ HAY QUE FILTRAR MORRALLA)

## LSB

<http://www.snotmonkey.com/work/school/405/methods.html>

Least significant bit (LSB) coding is the simplest way to embed information in a digital audio file. By substituting the least significant bit of each sampling point with a binary message, LSB coding allows for a large amount of data to be encoded. The following diagram illustrates how the message 'HEY' is encoded in a 16-bit CD quality sample using the LSB method:



In LSB coding, the ideal data transmission rate is 1 kbps per 1 kHZ. In some implementations of LSB coding, however, the two least significant bits of a sample are replaced with two message bits. This increases the amount of data that can be encoded but also increases the amount of resulting noise in the audio file as well. Thus, one should consider the signal content before deciding on the LSB operation to use. For example, a sound file that was recorded in a bustling subway station would mask low-bit encoding noise. On the other hand, the same noise would be audible in a sound file containing a piano solo.

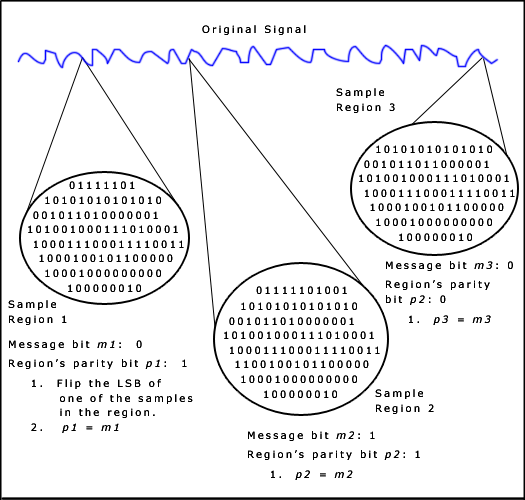
To extract a secret message from an LSB encoded sound file, the receiver needs access to the sequence of sample indices used in the embedding process. Normally, the length of the secret message to be encoded is smaller than the total number of samples in a sound file. One must decide then on how to choose the subset of samples that will contain the secret message and communicate that decision to the receiver. One trivial technique is to start at the beginning of the sound file and perform LSB coding until the message has been completely embedded, leaving the remaining samples unchanged. This creates a security problem, however in that the first part of the sound file will have different statistical properties than the second part of the sound file that was not modified. One solution to this problem is to pad the secret message with random bits so that the length of the message is equal to the total number of samples. Yet now the embedding process ends up changing far more samples than the transmission of the secret required. This increases the probability that a would-be attacker will suspect secret communication.

A more sophisticated approach is to use a pseudorandom number generator to spread the message over the sound file in a random manner. One popular approach is to use the random interval method, in which a secret key possessed by the sender is used as a seed in a pseudorandom number generator to create a random sequence of sample indices. The receiver also has access to the secret key and knowledge of the pseudorandom number generator, allowing the random sequence of sample indices to be reconstructed. Checks must be put in place, however, to prevent the pseudorandom number generator from generating the same sample index twice. If this happened, a collision would occur where a sample already modified with part of the message is modified again. The problem of collisions can be overcome by keeping track of all the samples that have already been used. Another approach is to calculate the subset of samples via a pseudorandom permutation of the entire set through the use of a secure hash function. This technique insures that the same index is never generated more than once.

## PARITY CODING

Instead of breaking a signal down into individual samples, the parity coding method breaks a signal down into separate regions of samples and encodes each bit from the secret message in a sample region's parity bit. If the parity bit of a selected region does not match the secret bit to be encoded, the process flips the LSB of one of the samples in the region. Thus, the sender has more of a choice in encoding the secret bit, and the signal can be changed in a more unobtrusive fashion.

Using the parity coding method, the first three bits of the message 'HEY' are encoded in the following figure. Even parity is desired.

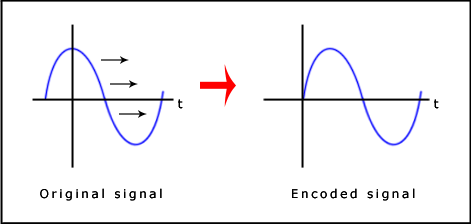


The decoding process extracts the secret message by calculating and lining up the parity bits of the regions used in the encoding process. Once again, the sender and receiver can use a shared secret key as a seed in a pseudorandom number generator to produce the same set of sample regions.

There are two main disadvantages associated with the use of methods like LSB coding or parity coding. The human ear is very sensitive and can often detect even the slightest bit of noise introduced into a sound file, although the parity coding method does come much closer to making the introduced noise inaudible. Both methods share a second disadvantage however, in that they are not robust. If a sound file embedded with a secret message using either LSB coding or parity coding was resampled, the embedded information would be lost. Robustness can be improved somewhat by using a redundancy technique while encoding the secret message. However, redundancy techniques reduce data transmission rate significantly.

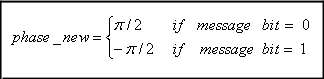
## PHASE CODING

Phase coding addresses the disadvantages of the noise-inducing methods of audio steganography. Phase coding relies on the fact that the phase components of sound are not as perceptible to the human ear as noise is. Rather than introducing perturbations, the technique encodes the message bits as phase shifts in the phase spectrum of a digital signal, achieving an inaudible encoding in terms of signal-to-perceived noise ratio.



Phase coding is explained in the following procedure:

1. The original sound signal is broken up into smaller segments whose lengths equal the size of the message to be encoded.
2. A Discrete Fourier Transform (DFT) is applied to each segment to create a matrix of the phases and Fourier transform magnitudes.
3. Phase differences between adjacent segments are calculated.
4. Phase shifts between consecutive segments are easily detected. In other words, the absolute phases of the segments can be changed but the relative phase differences between adjacent segments must be preserved. Therefore the secret message is only inserted in the phase vector of the first signal segment as follows:



5.A new phase matrix is created using the new phase of the first segment and the original phase differences.

6.Using the new phase matrix and original magnitude matrix, the sound signal is reconstructed by applying the inverse DFT and then concatenating the sound segments back together.

To extract the secret message from the sound file, the receiver must know the segment length. The receiver can then use the DFT to get the phases and extract the information.

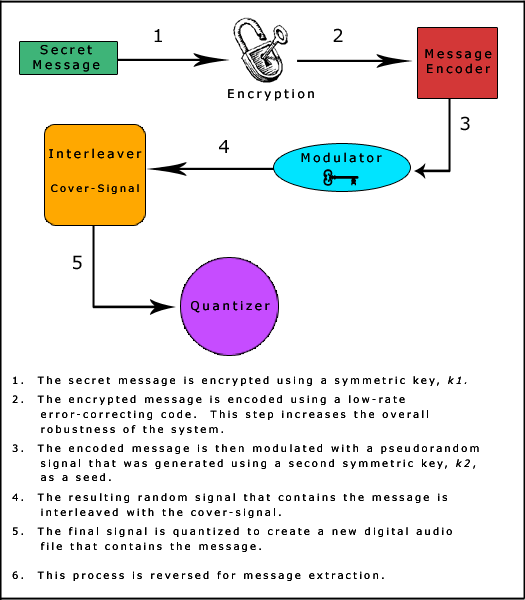
One disadvantage associated with phase coding is a low data transmission rate due to the fact that the secret message is encoded in the first signal segment only. This might be addressed by increasing the length of the signal segment. However, this would change phase relations between each frequency component of the segment more drastically, making the encoding easier to detect. As a result, the phase coding method is used when only a small amount of data, such as a watermark, needs to be concealed.

## SPREAD SPECTRUM

In the context of audio steganography, the basic spread spectrum (SS) method attempts to spread secret information across the audio signal's frequency spectrum as much as possible. This is analogous to a system using an implementation of the LSB coding that randomly spreads the message bits over the entire sound file. However, unlike LSB coding, the SS method spreads the secret message over the sound file's frequency spectrum, using a code that is independent of the actual signal. As a result, the final signal occupies a bandwidth in excess of what is actually required for transmission.

Two versions of SS can be used in audio steganography: the direct-sequence and frequency-hopping schemes. In direct-sequence SS, the secret message is spread out by a constant called the chip rate and then modulated with a pseudorandom signal. It is then interleaved with the cover-signal. In frequency-hopping SS, the audio file's frequency spectrum is altered so that it hops rapidly between frequencies.

The math theory behind SS is quite complicated and goes beyond the scope of this project. However, Katzenbeisser and Petitcolas write about a generic steganography system that uses direct-sequence SS in *Information Hiding Techniques for Steganography and Digital Watermarking*. The following procedural diagram illustrates the design of that system when applied to our specific topic of audio steganography.

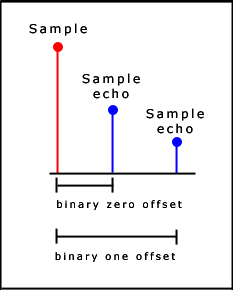


The SS method has the potential to perform better in some areas than LSB coding, parity coding, and phase coding techniques in that it offers a moderate data transmission rate while also maintaining a high level of robustness against removal techniques. However, the SS method shares a disadvantage with LSB and parity coding in that it can introduce noise into a sound file.

## ECHO HIDING

In echo hiding, information is embedded in a sound file by introducing an echo into the discrete signal. Like the spread spectrum method, it too provides advantages in that it allows for a high data transmission rate and provides superior robustness when compared to the noise inducing methods.

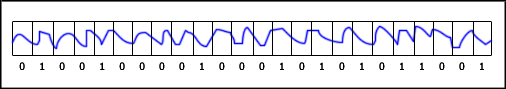
To hide the data successfully, three parameters of the echo are varied: amplitude, decay rate, and offset (delay time) from the original signal. All three parameters are set below the human hearing threshold so the echo is not easily resolved. In addition, offset is varied to represent the binary message to be encoded. One offset value represents a binary one, and a second offset value represents a binary zero.



If only one echo was produced from the original signal, only one bit of information could be encoded. Therefore, the original signal is broken down into blocks before the encoding process begins. Once the encoding process is completed, the blocks are concatenated back together to create the final signal.

We'll now go through a simple form of the echo hiding process using the message 'HEY'. For brevity, we'll divide the signal completely up into blocks, although under normal circumstances a random number of samples between each pair of blocks should remain unused to reduce the probability of detection.

First the signal is divided up into blocks, and each block is assigned a one or a zero based on the secret message. In this case, the message is the binary equivalent of 'HEY'.



Then the following algorithm (illustrated through pseudocode) is used to encode each block.

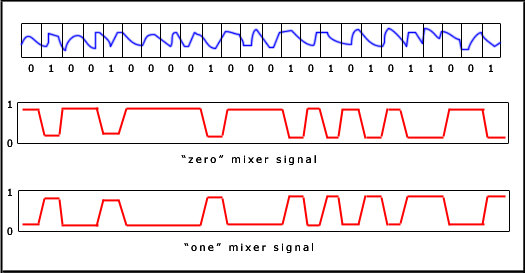
init(Block blocks[]) {   
   for (int i=0; i < blocks.length; i++) {   
      if (blocks[i].echoValue() == 0)   
         blocks[i] = offset0(blocks[i]);   
      else   
         blocks[i] = offset1(blocks[i]);   
   }   
}

Block offset0(Block block) {   
   return (block + (block - OFFSET\_0));   
}

Block offset1(Block block) {   
   return (block + (block - OFFSET\_1));   
}

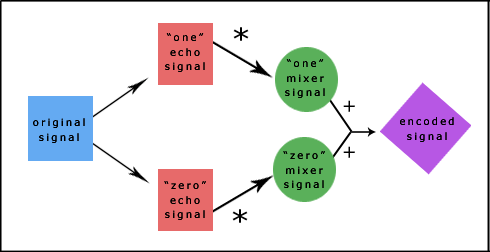
The blocks are recombined to produce the final signal.

Using that implementation of the echo hiding process can usually result in a signal that has a fairly noticeable mix of echoes, thus increasing the risk of detection. A second implementation of the echo hiding process addresses this problem. First an echo signal is created from the entire original signal using the binary zero offset value. Then a second echo signal is created from the entire original signal using the binary one offset value. Thus the "one" echo signal only contains ones, and the "zero" echo signal only contains zeros. To combine the two echoes together to get the final encoding, two mixer signals are used. The mixer signals have a value of either one or zero, depending on which bit is to be encoded in the block. In our example using the message 'HEY', we would get the following two mixer signals.



The "one" echo signal is then multiplied by the "one" mixer signal and the "zero" echo signal is multiplied by the "zero" mixer signal. Then the two results are added together to get the final signal. The final signal is less abrupt than the one obtained using the first echo hiding implementation. This is because the two mixer echoes are complements of each other and that ramp transitions are used within each signal. These two characteristics of the mixer signals produce smoother transitions between echoes.

The following diagram summarizes the second implementation of the echo hiding process.



To extract the secret message from the stego-signal, the receiver must be able to break up the signal into the same block sequence used during the encoding process. Then the autocorrelation function of the signal's cepstrum (the cepstrum is the Forward Fourier Transform of the signal's frequency spectrum) can be used to decode the message because it reveals a spike at each echo time offset, allowing the message to be reconstructed.