DIRECT-SEQUENCE SPREAD SPECTRUM METHOD APPLIED IN AUDIO STEGANOGRAPHY

Sergei Popkov

Summer School 2019 “Machine Learning for Speech”,
University of Eastern Finland,
rslw25@gmail.com

ABSTRACT

This work describes the DSSS (Direct-Sequence Spread Spectrum) method used to implement audio steganography task, where the password-protected arbitrary data is encoded into an audio file in a hidden way. The description of the DSSS method is provided along with the steps that implemented algorithm contains. Universality of the implementation is shown, as well as primary differences between LSB and DSSS methods. The purpose of all related files is described. The conclusion and results are provided, as well as possible future steps to advance the given solution.

1. SHORT TASK DESCRIPTION

Audio steganography: implement at least one other audio steganography method than the least significant bit (LSB) coding. Try to hide a text message or an image into audio file (file with the hidden message). Implement both embedding and retrieving of the secret message. Compare the implemented method(s) to LSB.

2. INTRODUCTION

The “steganography” term can be vaguely translated from Greek as “concealed writing” and is used to describe different methods and approaches that provide the ways to hide data in a cover media (“carrier”) in a manner that does not allow unaware people to notice it. While cryptography solves the task of protecting the hidden content, steganography hides the very fact of its presence and existence. Steganography can be applied in different areas (banking, military, medical and so on), and the cover media can be represented in different ways (audio, image or video). This work is focused on hiding data in digital audio files.

2.1. Steganography primary categories

There are six basic categories of different steganographic methods:

- Substitution methods replace unimportant parts of the carrier with a hidden data. The LSB method, for instance, belongs to this group.
- Transform domain methods work in a similar way to the substitution methods, but hide data in an audio frequency.
- Spread spectrum methods are based on spread spectrum telecommunication approach, making the carrier signal with hidden data to be transmitted on a larger bandwidth than the original information.
- Statistical methods encode the data by changing statistical properties of the carrier, testing hypothesis to perform the extraction process.
- Distortion methods store information using signal distortion.
- Cover generation methods perform the data hiding based on specific properties of the special purposely made cover format and data.

2.2. The reasons behind the category choice for this work

As a researcher, I found my interest, mostly, in the spread spectrum methods, as their basis came outside of the steganography itself, thus the domain of the spread spectrum is broader than steganography, making this knowledge more valuable as its products may be applicable in a broader way than the set task. From now onward this work is focused on the spread spectrum methods explicitly, unless stated otherwise.

3. METHODOLOGY

3.1. The principles behind the Spread Spectrum

A spread spectrum telecommunication is built on the principle of transmitting signals over wider bandwidth than strictly necessary to validly perform such operation. This ability is based on the information theory developed by Claude Shannon. This theory provides the formula for channel capacity:

$$Capacity = Bandwidth * \log_2(1 + SNR),$$

where SNR is the signal to noise ratio. The primary conclusion derived from this formula is that it’s possible to trade SNR to bandwidth and vice versa. This fact provides the means to encode the hidden data into a large signal bandwidth keeping transmission error-free under conditions where the noise is much more powerful than the signal.

3.2. The Spread Spectrum useful properties

The spread spectrum techniques are mostly used in GPS and military systems, where they're highly valued for the set of useful properties, namely:

- Bandwidth sharing and Code Division Multiple Access. It's possible to decode the same signal with different keys to extract different messages (CDMA).
- Security. It's hard to eavesdrop the message without the proper key.
- Stability. Any possible interference would be discarded with the rest of the noise.

There's two primary different approaches to the spread spectrum implementation: Direct Sequence and Frequency Hopping.

3.3. Direct Sequence Spread Spectrum (DSSS)

The DSSS method is used more frequently than FHSS because of its related simplicity yet practical usefullness. Direct sequence modulation is achieved by modulating the carrier wave with a sequence having much higher rate than the encoded message. Such sequence contains the pseudo-random code (also called "pseudo-noise", PN), that can be generated either based on a distribution with desirable statistical properties or as an additional protection layer (secure code or password). That way, the encoded message is transmitted within artificial noise, unknown to the unaware listener. The time period of a single bit transmitted along with the pseudo-noise is called a chip.

3.4. Frequency Hopping Spread Spectrum (FHSS)

The FHSS method is more complicated approach, where the frequency of the carrier is tuned to a pseudo-randomly determined frequency value. The spectrum is split up into a number of frequency channels. The carrier keeps popping up different frequencies in a pseudo-random pattern, allowing to modulate the cover media with the hidden message. Without knowing the code used for pseudo-random sequence generation, there's no way to determine the next used frequency channel, making the whole system much more resistant to any unwanted eavesdropping attempts.

3.5. Comparison of both methods; the choice of method for this work

The FHSS systems are harder to implement than DSSS systems. Nevertheless, the FHSS systems are more useful for dynamic audio sources, such as radio (static source, such as an audio file, can be analyzed for its frequency properties to extract the patterns of message encoding, as the frequencies remain static as well). DSSS, on the other hand, is easier and cheaper, resource-wise, to implement. It's more applicable to the static sources, such as audio files, as it's hard to recognize the noise as a potential hidden information source. As a result of the analysis, I find it more reasonable to implement the DSSS method in this work due to the nature of the task.

3.6. A short notice about windowing and signal smoothing

To avoid the spectral leakage (when the frequencies are leaked through each other; more detailed explanation of this phenomenon is beyond the scope of this work) and to make the pseudo-noice even less noticeable in spite of keeping the data spread all over the cover media, the practitioners in the field of communications recommend to smooth the encoded signal using the Hann smoothing (or Hanning window, which is available as Numpy embedded function). In short, the windowing procedure allows to ensure that the signal has a continuous waveform without sharp transitions, minimizing the spectral leakage. (This useful hint was provided by fellow scientists at the University of Communication and Informatics in my city).

3.7. Comparison with LSB (Least Significant Bit) method.

Spread spectrum implementations are more prone to carrier truncation and/or partial signal loss that LSB methods, because the hidden information is not concentrated in one place. Any noise interference in LSB-encoded cover media may lead to the disruption of the message, whereas the spread spectrum technique is more prone to the unwanted noise interference. LSB method is more easy to implement, though. Also, the LSB encoding does not relay on the signal reiteration and may allow to encode more information per bit. Nonetheless, the wise approach to the pseudo-noise generation allows the spread spectrum methods to share more than one message per signal, increasing its potential depth. Overall, the steganography system implementations based on the spread spectrum methods are more effective than the ones that rely on LSB approach.

4. IMPLEMENTATION & EXPERIMENTS

4.1. The general algorithm (message embedding)

1. Initialization: load the carrier, message and password.
2. Preparation: convert the message to the sequence of bits. Determine the chip (segment) length and quantity necessary to encode the message, mind bit alignment. Check if it's possible to store the message within the carrier without truncating, then proceed further.
3. Smoothing: apply the Hanning window to the sequence of bits (message) according to the chip length.
4. PN generation: create pseudo-noise bit sequence related to the provided password in a unique way. This sequence should have the same length as the segment. Propagate the sequence along the smoothed mixed signal. Check if the password is valid to apply the PN generation technique, update if necessary.
5. Message embedding: take the data from the first channel of the signal and add the value of multiplication to the first (L * N) values of the signal: (α * [Smoothed Mixed Signal Sequence] * [propagated PN]), where α is the embedding

strength, L is the length of the segment, N is the quantity of segments (the same as the count of characters to encode). This results in a pseudo-random noise carried along the cover media at higher frequencies than the media itself.

6. Finalization: return the modified signal data, password and the message length.

4.2. The general algorithm (message extracting)

1. Initialization: load the carrier, message length value and password.
2. Preparation: determine the segment length and quantity based on the message length, as well as the carrier length.
3. Message extracting (retrieving): for each segment, check average bit embedding influence considering the generated (based on provided password) PN noise and extract the expected bits encoding the message. For instance, this allows the random unwanted noise to be discarded, even for the unfortunate cases when some part of segment is damaged by the noise.
4. Finalization: return the message gathered from the extracted message bits.

4.3. Implementation “black box” (Input-Output) description

The embedding part takes as input the following parameters:

- The audio file name of the original carrier data (CARRIER_FILENAME)
- The audio file name of the resulting message-embedded carrier data (OUTPUT_FILENAME)
- The embedding strength (ALPHA)
- The text to embed (MESSAGE)
- The password for PN generation (PASSWORD)
- The smoothing parameter: lower bound of smoothed (mixed with the Hanning window) signal (SMOOTH_LOWER)
- The smoothing parameter: upper bound of smoothed (mixed with the Hanning window) signal (SMOOTH_UPPER)
- Hanning window parameter (number of points in the window) (SMOOTH_HANNING)

The embedding part returns as output the following results:

- The resulting audio file name
- The message length
- The (possibly corrected) password

The extracting part takes as input the following parameters:

- The audio file name of the carrier data containing the message to extract (STENO_FILENAME)
- The message length (MESSAGE_LENGTH)
- The password for PN generation (PASSWORD)

The extracting part returns as output the following results:

- The recovered message

Both extracting and embedding part share the constant:

- The minimal segment (chip) length (used to ensure the decent encoding quality) (L_MIN)

4.4. The implementation universality

The encoded data contains text message (not audio, image, video or any other media) with ASCII (8-bit) encoding of characters. However, this form of data is sufficient to carry any other media because of modern encoding schemes (such as base64) that provide the means to use an ASCII subset to represent any binary information. This fact proves the universality of the data encoding and format used in the current implementation.

4.5. Files used for an experiment

The following files are provided within the archive containing the work:

- DSSS.ipynb – The implementation (dependencies: Anaconda 3 with corresponding packages, librosa, soundfile).
- wave45.wav – The carrier input example, "Ambient Wave 45" by Erokia downloaded as a free resource from https://freesound.org/people/Erokia/sounds/482706/ (reencoded to reduce the size and be compatible with IPython).
- report.pdf – this document.

5. RESULTS

The message “The message to be hidden” was successfully embedded and retrieved from the carrier “Ambient Wave 45” with the embedding strength = 0.029, password = ”Tricky”. The noise is barely recognizable at a low sound volume, but is becoming more clear as the volume increases. The figure 1 contains the spectogram of original carrier signal before embedding, and the figure 2 shows the spectogram after embedding. The figure 3 demonstrates an example when the embedding strength is equal to 1, which eradicates the embedding/extraction errors, but the original sound is distorted by the noice to the point of being unrecognizable and disturbing to hear.

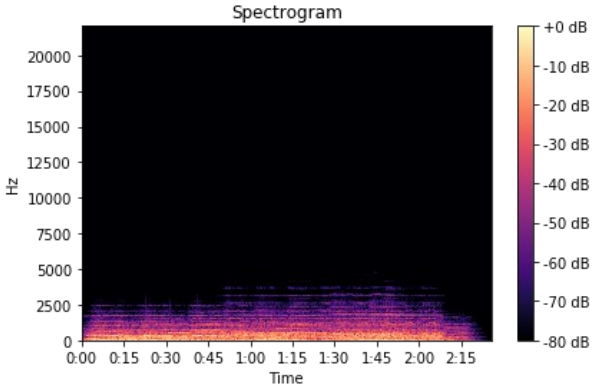

Figure 1. *Original carrier spectrogram (no embedding).*

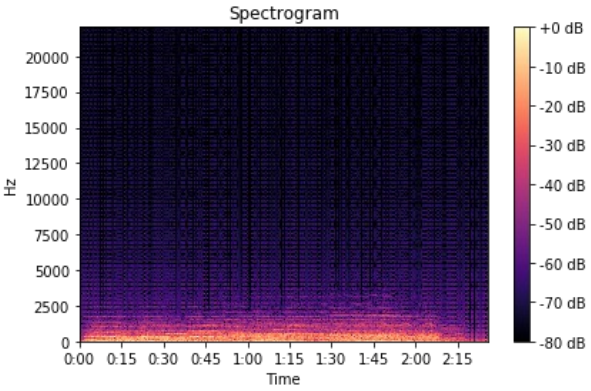

Figure 2. *Carrier spectrogram after embedding with embedding strength α = 0.029.*

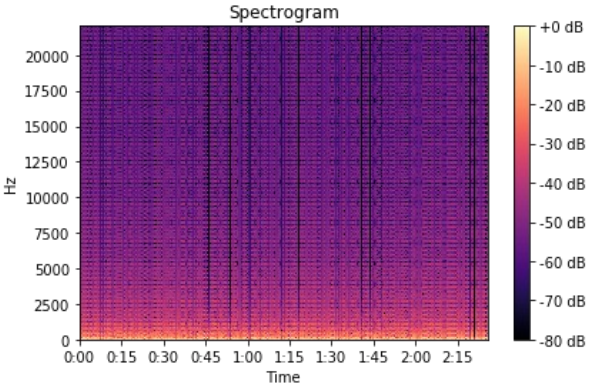

Figure 3. *Carrier spectrogram after embedding with embedding strength α = 1.*

5.1. Problems encountered during research

- The original audio file was not properly loaded by IPhyton. The problem probably should've been caused by the inner file format (encoding). Resaving the file via PySoundFile solved this problem.
- Lost the whole day trying to find any mistake in seemingly unworking code, but the problem actually was caused by the low value of the embedding strength, the program worked fine.
- The draft of the model was implemented in Excel first, where the slightly different value of embedding strength was satisfying for proper encoding than its implementation in Numpy. I assume this insignificant distinction between Excel and Python implementations could occur due to the different floating-point arithmetic (probably, precision).

6. CONCLUSION

The DSSS steganography method has been implemented. The implementation allows to encode the hidden message inside audio file with password which serves as a seed for pseudo-random PN sequence. The DSSS method is more stable than LSB method. The embedding strength parameter should be selected carefully, as its low value may lead to the message corruption, while too high value of the parameter can cause the significant sound distortion.

6.1. The future steps to study the problem further

- Implement FHSS, apply it to the learning video stream on YouTube (for example, to provide the way to extract the exclusive textbooks, slides and hints for clever students directly from the stream).
- Test how much information it is possible to encode in one-hour audio data using DSSS approach. Make a service to upload the data to the free audio hosting with auto split function (ability to upload more than one audio file containing the parts of the hidden file within).
- Apply various spread spectrum technologies to hide information within the image. Try to hide more than one file within one image using different PN codes.
- Think about possible cooperation between DAISY digital talking textbooks and DSSS encoding to enhance the experience of augmented reality for students with print disabilities (I teach some of them myself, too; this circumstance makes this task indirectly relevant to me).