

## Stéganographie : Cacher des messages dans du son

Voulant allier étude du signal et informatique, nous avons décidé de travailler au cryptage et au décryptage de messages dans des fichiers son, par la modification du codage de celui-ci.

Positionnements thématiques :

INFORMATIQUE (Informatique pratique), PHYSIQUE (Physique ondulatoire), SCIENCES INDUSTRIELLES (Traitement du signal)

Bibliographie commentée :

Pour coder des fichiers son, nous avons choisis de travailler en python, ce qui nécessite une prise en main de la bibliothèque wave [1], dédiée à la lecture et à la création de fichiers au format wave aussi connu avec l'extension « .wav ». Pour comprendre les différentes syntaxes et utilisations des objets pythons que nous avons utilisés, des tutoriels comme [2] par exemple, des documents explicatifs du format wave [4] ainsi que des exemples d'utilisation [5] nous ont donnés une base de connaissances pour écrire nos propres programmes ensuite. Dans un but de vérification rapide des modifications effectuées par les programmes, nous avons utilisés un éditeur hexadécimal [6].

Cependant, pour pouvoir décoder le message caché dans le son suite à une diffusion, un enregistrement, et un traitement, nous sommes confrontées aux phénomènes de bruit. Ainsi nous souhaiterions explorer plusieurs pistes comme du codage sur des bits de poids plus fort, ou si possible l'adaptation de la segmentation voix/musique [3], à un filtrage du son séparant musique et bruit.

Problématique :

Il s'agit d'étudier les limites du cryptage par changement de l'encodage d'un fichier son, c'est à dire dans quelle mesure des modifications de fichiers restent inaudibles tout en étant utilisables en cryptographie.

Objectifs pour la fin d'année :

- trouver les différents seuils acceptables pour coder un message dans un fichier sonore, sans que cela provoque une différence notable à l'oreille
- ces limites une fois trouvées, chercher des moyens pour accéder au message caché, de façon uniquement informatique dans un premier temps, puis suite à une diffusion et une captation, en minimisant le bruit ou en filtrant informatiquement l'enregistrement obtenu.

Références bibliographiques :

- [1] Python Software Foundation, documentation pour le traitement de fichiers wave, <https://python.readthedocs.io/en/stable/library/wave.html>
- [2] Jean-Luc DANIAU, écrire un fichier son avec Python, <https://www.youtube.com/watch?v=67Qt6q8u8Fs>
- [3] Theodoros Giannakopoulos, pyAudioAnalysis: An Open-Source Python Library for Audio Signal Analysis, 2015, PLoS ONE 10(12) : e0144610. doi : 10.1371/journal.pone.0144610
- [4] Wikimedia Foundation, forme des fichiers de forme WAVE [https://fr.wikipedia.org/wiki/Waveform\\_Audio\\_File\\_Format](https://fr.wikipedia.org/wiki/Waveform_Audio_File_Format)
- [5] GeeksforGeeks, fonctionnement de « 'rb' » et des « bytes object » en python 3, <https://www.geeksforgeeks.org/reading-binary-files-in-python/>
- [6] Jens Duttke, éditeur hexadécimal, <https://hexed.it>