

Práctica 5: Esteganografía 1

Método LSB

- 1 Revisar la documentación para la librería Pillow para manejo de imágenes. Ver qué comandos permiten traducir una imagen a una matriz ya sea de tres parámetros (grayscale) o 3-dimensional (RGB).
- 2 Construir las funciones **texttobit** y **bittotext** que permita pasar un texto a su equivalente en bits (previo paso a ascii) y viceversa.
- 3 Construir una función **LSBsimplecypher** que tome un texto y una imagen y devuelva una imagen en blanco y negro con el texto cifrado en ella. Se deberán utilizar los primeros píxeles para esconder el mensaje. También será necesario determinar si la imagen es suficientemente grande para ocultar el mensaje entero. De igual forma, construir **LSBsimpledecypher** que tome una imagen en blanco y negro y devuelva el mensaje codificado en él.
- 4 Construir una función **LSBcomplexcypher** que tome un texto y una imagen y nuevamente devuelva una imagen en blanco y negro con el texto cifrado en ella. En este caso, el programa deberá sugerir valores s que permitan esconder el mensaje usando, no los primeros píxeles, sino aquellos obtenidos al hacer un salto de s -píxeles (el primer número se oculta en el píxel s , el segundo en $2s$, etc.) De igual forma definir **LSBcomplexdecypher** que tome una imagen en blanco y negro y un valor s y descifre el mensaje.

Desordenando una imagen

- 1 Defina una función **isinvertible** que tome una matriz cuadrada 2×2 y un entero n y determine si la matriz es invertible en \mathbb{Z}_n .
- 2 Defina una función **powinverse** que tome una matriz A y un escalar n y determine cual es el primer p de modo que $A^p = I$. De este modo, $A^{p-1} = A^{-1}$.
- 3 Defina una función **desordenaimagen** que tome una matriz A y una imagen, compruebe si A puede utilizarse para desordenar la imagen y devuelva la imagen desordenada (ya sea grayscale o RGB). De igual forma, defina apropiadamente **ordenaimagen**.
- 4 Defina una función **desordenaimagenite** que realice lo mismo que antes, pero que además le pida al usuario un valor k de modo que la imagen se desordene no con A sino con A^k . El programa debe indicar qué k son adecuados para calcular A^k . De forma adecuada defina **ordenaimagenite**.
- 5 Finalmente, crear un programa **desordenaimagenproceso** que utilice **desordenaimagenite** de forma iterada para ver la evolución de la imagen con los distintos valores de k . Os dejo libertad para implementar el número de imágenes que debe mostrar el programa para ver la evolución.

Recordad que todo el código realizado deberá estar convenientemente documentado.