

Esteganografía

Joaquín Pérez Araya
Departamento de Ciencias de la Computación
Universidad de Chile
Santiago, Chile
joaquin.perez.a@ug.uchile.cl

Resumen—

INTRODUCCIÓN

A lo largo de la historia de la humanidad ha existido la necesidad de enviar información ocultos, ya sea para transmitir información sensible, como Ejemplos... Uno de los modos es a través de la Esteganografía, que viene de ... En este documento se mencionará una implementación de Esteganografía sobre imágenes ... que consiste en ... la ... ejemplos de Esteganografía en imágenes (usar [1]) más específicamente un método bastante simple que consiste en... (tablita, algoritmo bonito, blablabla)

Dentro del texto actual

DISEÑO E IMPLEMENTACIÓN

Para la implementación del programa se utilizó Python 3.6 con las librerías `numpy` y `scimage` y `argparse`. El código está dividido en 4 partes: Utilidades, Entrada/Salida, Codificación y Decodificación.

Entrada y Salida

Para el uso externo del programa por medio de comandos se utilizó la librería `argparse` que viene por defecto en Python. Leer los comandos dados y traducirlos a llamados directos a las funciones de codificación o decodificación.

Utilidades

El módulo de utilidades (`util.py`) están las funciones auxiliares que utilizan la codificación y decodificación entre las de las cuales se destacan:

- `image_read`¹, `image_write` y `text_read`: Son las funciones para leer los archivos externos que se van a utilizar para el proceso de codificación y decodificación.
- `text_to_ascii` y `ascii_to_text`: La primera se utiliza para codificar el mensaje mientras que la segunda para decodificar. Se usan para transformar una cadena de texto a una lista de números donde cada carácter es un número de la lista y viceversa.
- `to_binary` y `to_int`: Se utilizan para codificar y decodificar, ya que reduce el problema de inserción de bits a modificar las cadenas de texto en binario de los valores del canal de imagen.

¹La función que está implementada en el código es la que está en `pai_io.py` dentro del repositorio del curso.

- `join_binaries`: Se usa para modificar la cadena de texto binaria para codificar un carácter en éste.
- `last_value`: Se utiliza para obtener los bits menos significativos de un número, para obtener el carácter que está codificado es éste.

Codificación

La codificación consta de una única función que dada la una dirección de imagen, una dirección de texto y un número de bits realice todo el proceso de abrir la imagen, obtener el primer canal de información para modificarlo con la finalidad de agregarle la información que corresponde al texto. Dado el primer canal de la imagen dada, se codifica en los cuatro bits menos significativos el número de bits que se van a usar en los siguientes píxeles para facilitar la acción de decodificación. Posteriormente, se convierte el texto a codificar a una lista de números donde letra se le asigna un número que corresponde al valor en ASCII del carácter.

Dado a que cada símbolo en ASCII es un número de 8 bits, en cada píxel se puede elegir la cantidad de bits a codificar, éste método permite codificar $\lfloor (p-1)(bits/8) \rfloor - 1$ caracteres como máximo dentro de una imagen con p píxeles en total, ya que en el primer píxel de la imagen siempre se va a utilizar para guardar la cantidad de bits utilizado en los 4 bits menos significativos de éste, también se necesita codificar un carácter adicional para indicar que no se debe seguir decodificando.

EXPERIMENTACIÓN

Inicialmente para el testeo del funcionamiento inicial de la implementación se utilizó una imagen en negro de 10×10 píxeles con la finalidad de verificar vía simple inspección la codificación/decodificación.

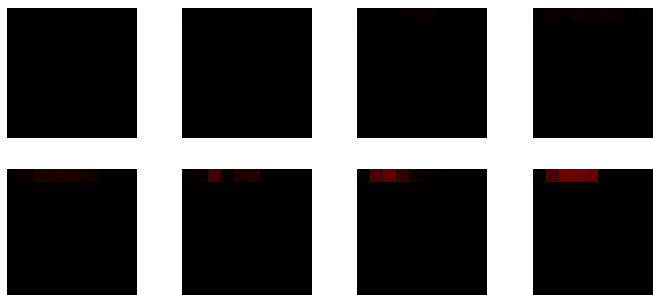


Figura 1. El texto `Hola`, codificado usando 1 a 8 bits.

CONCLUSIÓN

Muy bien me gusto, pongame un 7.0 tkm

REFERENCIAS

- [1] Cheddad, A., Condell, J., Curran, K., & Mc Kevitt, P. (2010). Digital image steganography: Survey and analysis of current methods. *Signal Processing*, 90(3), 727–752.