Report of ECI 2019 Course: “ Introduction to Steganography and Watermarking “

Assignment E.316-N

Acha Francisco, Caldo Juan Pablo, Cardenas Rodrigo, Castagna Franco and Remedi Elias.

1 Introduction

Steganography is the procedure of insert information inside a data source without changing its perceptual quality. Digital steganography uses digital data sources as a cover for hidden information. Examples of digital covers are digital text files, image files and sound files among others.

In particular for digital image based steganography the pixel intensity is usually used for encoding information [REF] but other approaches are also widely used such as embedding information in the frequency domain.

There are available many software tools

In this report, five steganographic tools that hides text into a digital image were chosen to perform an assessment in terms of imperceptibility of the stego-image, capacity and robustness.

2 Materials and methods

2.1 Dataset

Since we want to evaluate performance of steganographic tools that hide text into an image, a image dataset is needed. We built a dataset containg images 20 of four types: N-type (landscapes and open nature), S-type (still life), P-type (portraits) and T-type (text). The complete dataset is then 80 images in total. N, S, P-type images were obtained and selected from Google images search engine queries. Namely, keywords for queries were *landscapes*, *still life* and *portrait* respectively. Right usage for the images was selected such that results were labeled for noncommercial reuse, and size of the images was set in medium [NOTA AL PIE DE LA FECHA]. Text images were collected from research papers by exporting pages as jpeg images. Table [REF] summarizes some basic features of the dataset used such as mean image size and mean file size. All the images in the dataset were stored as jpeg format. DECIR AHORA EL TAMAÑO MEDIO, Y LA MEMORIA DE CADA IMAGEN MOSTRAR UN EJEMPLO DE CADA TIPO

2.2 Description of selected software

2.2.1 *OutGuess (v. 0.2)*

OutGuess is a universal steganographic tool that allows the insertion of hidden information into the redundant bits of data sources. The nature of the data source is irrelevant to the core of outguess. The program relies on data specific handlers that will extract redundant bits and write them back after modification. Currently only the PPM, PNM, and JPEG image formats are supported, although outguess could use any kind of data, as long as a handler were provided. *OutGuess* uses a generic iterator object to select which bits in the data should be modified. A seed can be used to modify the behavior of the iterator. It is embedded in the data along with the rest of the message. By altering the seed, outguess tries to find a sequence of bits that minimizes the number of changes in the data that have to be made.

A bias is introduced that favors the modification of bits that were extracted from a high value, and tries to avoid the modification of bits that were extracted from a low value.

*A*dditionally, OutGuess allows for the hiding of two distinct messages in the data, thus providing plausible deniablity. It keeps track of the bits that have been modified previously and locks them. A (23,12,7) Golay code is used for error correction to tolerate collisions on locked bits. Artifical errors are introduced to avoid modifying bits that have a high bias.

2.2.2

2.2.3 *StegHide (v. 0.5.1)*

StegHide is an open source steganographic software that allows hide text using image or sound files as covers. (REF A LA PAGINA) It supports JPEG, BMP, WAV and AU file formats as cover files. *StegHide* performs steganography by means of a graph-theoretic approach. Data to be embedded is compressed and encrypted, Then a pseudo-random sequence of postions of pixels is created. On this positions secret data will be embedded. Then a graph-theoretic matching algorithm finds pairs of positions on the remaining pixels such that exchanging their values has the effect of embedding the corresponding part of the secret data. If there are not enough pixels with values that can be used to embed the data by exchanging, values are overwroute. This way, most of the embedding is done by exchanging pixel values and then the first-order statistics is marginally changed. A passphrase must be provided by the user for encryption and pseudo-random generator initialization. The same passphrase must be provided for data extraction from stego-file. The default encryption algorithm is Rijndael with a key of 128 bits altough others are available as well.

2.2.4

2.2.5 *SteganPEG*

2.3 Benchmarking

Some criteria and metrics needs to be established in order to to benchmark the selected software. In this section metrics for imperceptibility assessment are presented as well as criteria regarding capacity of storage for hidden data and tests for robustness evaluation.

2.3.1 Imperceptibility

Imperceptibility is a measure of how much the stego-file looks like the original cover file. Is a critical attribute of a steganography system. Having a poor imperceptibility level will make it easier for an eavesdropper to detect the presence of a secret message. For imperceptibility assessment we have selected two widely known metrics namely, Peak Signal to Noise Ratio (PSNR) and Structural Similarity (SSIM).

2.3.2 Capacity

Capacity refers to the size of the hidden secret data that a steganographic system is able to embed into to the cover medium. Capacity of a steganographic system depends on the method used to hide data. Hence, different systems can embed different amount of data given the same cover. Capacity for digital image-based steganograpy can be computed as (PONER ECUACIÓN) [REF]

2.3.3 Robustness

Robustness of a steganographic medium is the property of keep the secret information undestroyed under transformation or tampering of the stego-file. Sistematic assessment of robustness can be done by applying known transformations to the stego-file and evaluating the quality of the hidden data recovered from it. In this work robustness for each steganograpic tool was tested under five tampering scenarios: overwriting of 10% of pixels to zero; add of gaussian noise with $\mu = 0$ and $\sigma^2 = 2.5\ 10^{-3}$; horizontal stretching by 30%; vertical mirroring; and 20^o counter clock-wise rotation.

3 Results

3.1 Imperceptibility

3.2 Capacity

3.3 Robustness

4 Discussion and Conclusion

5 References