Steganography

7 Steganography - Hiding Data in Plain Sight Steganography is the art of hiding data in plain sight. Security of files can be achieved through encryption, which is the process of manipulating the data within the file, in a manner derived from a key. Encrypted data though is easily identified, as it has many statistical properties, such as uniformity of the distribution of the data within the file. As this data is very uniform, properly encrypted data is therefore indistinguishable from random data. Sometimes it is desirable to mask the fact that encryption is in use, or instead to simply store data without it being obvious. The technique of steganography was first used in the days of the Roman Empire (around 440 BC), where messages needed to be transmitted, safely, over long distances, through enemy-held territory. If the messenger was stopped by enemy forces, it would easily be discovered and the content (often battle plans or information about observations of enemy movements) would be known to the enemy. To prevent this from happening, the message needed to be hidden; a messenger would have his head shaved, and the message would be tattooed to his scalp, in the colour of his hair. The messenger would then grow his hair, and cross the enemy territory, undetected. If searched, no message would be found. At the other end, the messenger would shave his head, and reveal the message to the recipient. A modern-day use of steganography is to hide data within a standard (and inconspicuous) file, such as an image or audio file. The file should continue to appear (or sound) correctly, such that the hidden data would not be spotted. The data should be difficult to spot, even when looking for it (the equivalent of the searching of the messenger) good steganography will result in the data being invisible to even skilled attempts to find it. Hiding data uniformly throughout the file should help to achieve this. Carefully hiding the data within the file will result in data which is more difficult to identify. Try to avoid introducing significant patterns of variation of the file as a result of adding the hidden data.

**Personal Section**

1. Understand some techniques which can be used to hide data in files.

**2. Research popular file formats for images and sound, and identify those which are ideal to hide data within.**

3. Write a program to hide data (a file, or user input) within a user-selected image or sound file.

**4. Write a program to retrieve hidden data from a selected image, if it exists, and output it to a file or text.**

5. (Optional) Implement encryption of the data hidden within the file.

**6. (Optional) Attempt to write code to detect the presence of your hidden data within a file.**

<http://security.stackexchange.com/questions/2144/detecting-steganography-in-images>

<https://en.wikipedia.org/wiki/Steganography#Citations>

<https://en.wikipedia.org/wiki/Steganalysis>

<https://en.wikipedia.org/wiki/Flowchart>

<https://en.wikipedia.org/wiki/Pseudocode>

<https://en.wikipedia.org/wiki/Software_testing>

<http://www.python-course.eu/tkinter_layout_management.php>

<http://usingpython.com/using-tkinter/>

<https://www.packtpub.com/mapt/book/networking-and-servers/9781784392932/6/ch06lvl1sec52/hiding-a-message-using-lsb-steganography>

<https://docs.python.org/3.0/library/csv.html>

<https://pythonprogramming.net/change-show-new-frame-tkinter/>