CT255 Assignment 4

Steganography

Overview

The objectives of this assignment are as follow:

1. Reinforce your understanding of steganographic techniques.
2. Implement a simple steganographic algorithm.
3. Improve your implemented algorithm.

The Java class *Stegano1* contains the skeleton code for a steganography class, including the command line interface we've used in previous assignments. The latter accepts command line arguments as shown in the following examples:

1. Adding a bit vector to a file:	Stegano1 A inp.txt out.txt 0010101
2. Retrieving a bit vector from a file:	Stegano1 E inp.txt

Fully implemented (by yourself), the code will **A**dd a secret bit vector / message (e.g., "**0010101**") to an existing text document ("**inp.txt**" above) and saves the resulting text under a different file name ("**out.txt**" above).
Similarly, a hidden bit vector can be **E**xtracted from a file and printed on screen using the 2nd example.

Problem 1: [5 marks]
Complete the code by implementing the following "hiding" algorithm:

- Store one bit of the input vector per line of text you are processing; as you can see from the Java code provided, the *hide()* method reads one line of text at a time:
  - For hiding a "0", a single space (" ") will be appended to the end of the line.
  - For hiding a "1", two spaces (" ") will be appended to the end of the line.
- Retrieving a secret message works accordingly.

Demonstrate how your solution works by hiding your first name (encoded as 8-bit ASCII) in the attached text file ("The Stolen Child" by W.B. Yeats), and subsequently retrieving it.

Please note:

- The secret message / bit vector is stored as a string object, i.e., your solution is basically a series of string operations using methods already shown / exercised in previous assignments. All the file I/O operations and the command line interface are already provided by the skeleton code.
- Make sure that you consider boundary cases, i.e.:

    - The secret bit vector must fit into your file, e.g., if your text is only 3 lines long, you cannot hide more than 3 bits in it.
    - Message retrieval of a normal text file (i.e., a text without added spaces) will return nothing (as you would expect).

Problem 2: [5 marks]
Enhance your code in problem 1 to store <u>two</u> bits per line of text. Fully explain your solution in the source code. Further on, demonstrate how your solution works by hiding and retrieving your surname encoded as 8-bit ASCII, again using the attached text file.

Please note (on top of my previous comments in problem 1):

- Append a padding bit ("0") to the end of your input bit vector if it has an odd length.
- Think of a good way to hide the additional bit, e.g., adding spaces at the beginning of a text line is too obvious and defeats the purpose of steganography!

Assignment Submission

Please submit a zipped folder to Canvas containing:

- Your (well-commented!) source code for problems 1 and 2 in PDF format.
- Screenshots showing your programs being compiled and producing results.

Marking Scheme

For both Problem 1 and Problem 2:

- 4 marks for a working and well documented solution, 1 mark for attempt.
- 1 mark for demonstrating the hiding and retrieving your name via a series of screenshots.