Advanced Computer Security  
Assignment 2 (W18)

– Digital Camera/Steganography Lab

**DUE: Consult eConestoga’s ACS Schedule or DropBox for the current due date of this assignment.**

**Note: You may work on your own or with a partner (recommended) for this assignment**

Overview

This assignment has two main components. First, students are required to write a utility application that can read through a disk image file from a camera’s SD card, retrieving full and partial images. Once images have been extracted, they must determine some way to encode a simple message back into the files. The primary products of this assignment are the code for the utility application and a results document that outlines the student(s)’ approach to and success with the assignment.

Scenario and Requirements

I’ve prepared two disk image files you will find in eConestoga along with this instruction document. There is a ‘control’ file and a file marked ‘wiped’. These are provided for your testing purposes, and were made on an older (circa 2000-2001) HP digital camera with a 4MB SD card. The main purpose of the assignment is for you to find every picture you can from the ‘wiped’ disk image. Here, like every other normal user, I’ve used the camera’s tools to wipe the SD card, so there should be nothing to find, right?

1. Develop an application that will scan through the provided disk images files.
   1. To help you, there are many resources online that will be of help, but the instructor will also prepare two versions of the camera’s SD image – before and after deletion.
   2. Use the two provided disk image files to test your application
2. Once your application can recover the image files, take some screen shots of your results for use in your assignment document. Better yet, dump the images you find (whole and partial) to a zip file for my review along with the other deliverables.
3. Next, your application needs to embed a secret message back into one or more of the image files you recovered. This is called steganography. **You MUST hide your messages using a bit-manipulation technique like least significant bit… this may require some file conversion! Hiding your message in plain sight (e.g. within a file’s meta data header or such) is not permitted… you need to get crafty!**
4. Once the message is embedded in the image file, show that the image file can be displayed without any untoward artifacts appearing (e.g. prove the message is hidden).
5. Demonstrate how your application can read the hidden message from the image file. This is somewhat similar to the requirement in the Crypto assignment that we can verify results.

# Your deliverables

Create a results document that will describe your approach to the assignment, and show the files that you retrieved, and how your hidden message is embedded/hidden as per the notes above.

**Other Code Bases and Referencing** - Although there are a lot of tools and tutorials on recovering deleted files and embedding messages within files, you are expected to produce your own application for the assignment and to fully cite your sources where you borrow an approach or some insight.

The two main areas where you are expected to create your own code to solve a challenge are a) parsing the data from the SD card images to look for and extract full and partially deleted image files, and b) manipulating a chosen image to record a hidden message (steganography).

You might find that certain techniques covered in other courses can ease your development work, and you might have some easy to deploy UI for a project like this. If you have previously developed code that could be repurposed to help you in this assignment I encourage this, and simply ask that you properly cite your previous code base at within code comments and within your results document.

## What to Hand In:

Via eConestoga drop box, in a zip file, please provide:

* Your working, well commented, recovery and message-embedding VS solution
* Results document – see above for the content I’d like to see here.
* (optional) a folder containing copies of the images you recovered from the wiped disk image.

## Marking Guidelines

Application can pull full and partial images from the wiped disk file – 10pts

Application has a module allowing for the hiding of text messages within a recovered image file – 10pts

Results document (formatting, layout, citing resources, professionalism, etc) - 10 pts

**Total – 30 pts**