Murder Mystery Lab

Instructor Guide

- Overview
- Learning Goals
- Personal Growth Goals
- Skills Required
- Resources Required
- Instructor Preparation
- In-Depth Description
  - Step 1: Decoding the Hint
  - Step 1.5: Optional Activities/Discussion
  - Step 2: Steganography
  - Step 2.5: Optional Activity/Discussion
- Lesson Plan

Overview

This lab covers alternative ways of representing data. Previously, we learned the standard way computers store data, which is using binary numbers: 0s and 1s. However, what if you want to send data in secret, so no one, not even the computer, can know what you are sending? This can be achieved through cryptography, which means encoding and decoding messages so others cannot read or see their contents.

This whole lab is based on a murder-detective scenario, which the instructors will explain to the class beforehand. The students all play detectives, whose job is to investigate the murder of Mrs. Larkin, director of a large law firm, Larkin Law. She was found dead in her Pittsburgh penthouse at 8:00 PM on Wednesday night, when her husband, a police officer, came home from work. As the detectives looked for evidence, they found something strange. Six seemingly-random images of black-and-white squares were hidden in her bedroom! They knew something was going on, and decided to send the images to the lab to investigate.

Learning Goals

- To reinforce students' understanding of how numbers and strings are represented in computers
- To reinforce students' understanding of loops and 2D lists
- To teach students alternative ways to represent data (as opposed to the standard ways discussed in the previous lab) and why it is useful.

Personal Growth Goals

- How to search the internet, both for fun facts and for technical help
- Patience: this lab has a small group component, but then students have to wait for other groups to get done. Students will have to exercise patience, and also explore what to do in those situations (i.e. help the other groups, work on or think about more advanced topics, etc.)

Skills Required

- An introductory understanding of functions, loops, and conditionals ()
- An understanding of how numbers are represented in computers (binary)
- An understanding of how images are stored in computers (a 2D array of RGB values)
- An understanding of lists and indexing into lists

Resources Required

- One mentor per student or per 2 students.
- One internet-connected computer per mentor
- One smartphone per mentor

Instructor Preparation

1. Load steganographyHelpers.py onto each computer.
2. Load a QR Code Reader onto every mentor's phone.

There is already a sample of the necessary hints for this lab located here. This sample is made for 6 groups (ideally pairs) of students. If instructors want to change the resources in the sample, these are the steps to generate them:

1. Decide on the number of groups students will be divided in (ideally students will be in pairs.) This number will be referred to as n later in the lab.
2. Choose the image to encode the final message in, and the image to hide. If you want to add text to the images, you can use PicFont. The images must be GIFs, so if need be use Convert To Gif.
3. Use the following Instructor Code to hide the private image in the public image. You may have to play around with the reduction factors a bit, and may even have to make the private image black and white in order to encode it (GIF allows max 256 unique colors.)
4. Upload the encoded image to the Teknowledge website.
5. Divide the url up into n about-equal parts (if we need to make the url longer, we can give the encoded image a long, semi-random name). For each part, decide on how you will encode it (examples: Caesar Cipher, Substitution Cipher, Transposition Cipher,

reversing, some combination thereof, etc.) Then write instructions for the students about how to decode the message, ideally involving some online searching. Also include which part of the full message they have. An example of a complete string (taken from the Sample) is as follows:

> 1) *How are letters represented in a computer? (Hint: find out what chr and ord do)*
> 2) *On what date (the day number) did Beyoncé release the YouTube video for Halo?*
> 3) *Using a trick called a Caesar Cipher, I hid a message in the QR code. I converted each letter to a number, shifted it up by the number you found in the previous step, and converted it back to a letter. Can you decode what I wrote?*
>
> *This is the first part of the message. The other QR codes have the other parts, feel free to help debug those until all are solved: kwwsv=22jlwkxe1frp2Wh*

6. Take each hint (a string similar to the string above) and convert it to a QR code using QR Code Generator. Then print the QR codes out

In-Depth Description

The mentors will start by giving each group of students a QR code as a hint. If the students recognize it as a QR code and realize that you can get an app to read it, then let them read it and proceed with the lab. If not, mentors can walk the students through what a QR code is, and how it is a commonly used for of encoding various data types into an image. The students will then scan in their QR codes, and see the hint.

Step 1: Decoding the Hint

The sample hints have a couple of different steps:

1. Search the internet for how to do something in Python (ideal places to get this information are Python Documentation or StackOverflow). For this step, the mentors should work with the students to understand the value of the internet as a way for getting coding help, and discuss credible sources and how to find them.
2. Search the internet for some piece of trivia. This again teaches students internet searching skills and will reinforce the belief that sometimes knowing is not as important as knowing how to learn it.
3. Write code to decode the string. This step involves:
    a. Reading and understanding how Mrs. Larkin encoded the message,
    b. Reversing that logic in order to decode it, and
    c. Translating that into code.

Since some of the code students will have to write involves functions or ideas that they haven't learned yet, mentors will have to work closely with students to decode the message.

In the sample, Mrs. Larkin uses three kinds of encoding: Caesar Cipher, Substitution Cipher, and Transposition Cipher.

For all the above steps, there is no clear way to divide up the work amongst members of the groups. One suggested allocation (assuming groups of two), is: one students does #1 and one student does #2, and for #3 both students participate in interpretation and idea generation, one writes the code and the other suggests what to write.

Step 1.5: Optional Activities/Discussion

As the students realize, the message that they just decoded is only part of the whole message. If they want to, they can help other groups decode their message. If not, this is a good time for mentors to discuss any of the following:

1. How QR codes work,
2. The uses of ciphers,
3. Other ciphers the students can think of (maybe they can code them up?), or
4. The Enigma cipher and its importance in WW2 and the field of CS

Step 2: Steganography

After all student groups have decoded their string, they will realize that it is a url. When they go to that url, they will see the second clue, an image, with brief instructions about how to decode it. Since the instructions are brief, the mentors may have to explain more about steganography and how it works. The sample image was made by iteratively taking the (r0,g0,b0) values of the private image and hiding them in the (r1,g1,b1) values of the public image by making the last bit of each color in the public image 1 if the corresponding color in the private image is 128, else 0. (Before this we reduced the colors in the private image so each (r,g,b) value is either 0 or 128.) This way, you minimally change the public image (changes of 1 unit per pixel are not discernible by the human eye), and you can convey all information about the private image with just one encoded image.

In order to find the private image, students will write their own steganography decoder function. steganographyHelpers.py has functions to convert a GIF image into a 2D list of (r,g,b) 3-tuples and to write a 2D list of (r,g,b) 3-tuples to a GIF file. All the student needs to write is the code to iterate over these RGB values, extract the information about the hidden image, add it to a new 2D list, and write that 2D list. Mentors should work closely with the students to write this code.

Once the students have decoded the image, they have solved the mystery! Congratulations! Students are welcome to use any of the ciphers learned in this class or steganography to pass notes in the class moving forward :)

Step 2.5: Optional Activity/Discussion

If the students finish early, mentors can discuss any of the following with their students:

1. Google Deep Dream. The sample public image is from Google Deep Dream, and students might be interesting in how it creates these awesome superimpositions of images solely through computer algorithms.
2. How would the students go about writing a steganography encoder? If the students can write code for it (all necessary helpers are included in steganographyHelpers.py) that would be great! Mentors should also explain to students GIF's 256 unique color limit, and how that affects how they reduce colors in the public and private images.
3. Once the students go through the process of designing or coding a steganography encoded, there are a number of follow up questions the mentors can ask:
    a. If instead of hiding an image inside another image, you want to hide a string inside an image, how might you do that?
    b. If instead of hiding one image inside another, you want to overlay two images, how might you do that? (One possible answer: average them)

Lesson Plan

(:10) means that this part should be done by the tenth minute of the lesson

1. Setup (:10)
2. Step 1: Decoding the Hint (:40)
3. Step 2: Steganography (:90)