Whatcom COMMUNITY COLLEGE

NCYTE CENTER

STEGANOGRAPHY

Sean Gallop – Colorado Academy, Boulder, CO

This material is based upon work supported by
the **National Science Foundation** under Grant No.1548315.

Additional materials may be found at www.ncyte.net

STEGANOGRAPHY LESSON

OVERVIEW

Steganography is the process of hiding information inside of a file, message, photograph, or video. Steganography can appear in many forms, but, typically, steganography consists of hiding a secret text message or photograph inside of a digital photograph. The photograph included immediately below appears to be a picture of a tree. However, the image on the right contains a hidden message. Alternatively, steganography can consist of hiding video inside of a sound file. This version of steganography figures prominently in the movie Contact (1997) in which aliens embed a video into their first radio wave communication with Earth. Steganography can also consist of an image concealed inside of another photographic image. This version of steganography, called autostereograms, was popularized by the Magic Eye® image craze of the 1990s in which a three-dimensional image was hidden inside of two-dimensional photographs.

Steganography extends the typical definition of cCryptography. Whereas cryptography is concerned primarily with the protection of secret (encoded) messages. Steganography is concerned both with the protection of secret messages as well as the means by which those secret messages are transferred.

This document is licensed with a Creative Commons Attribution 4.0 International License ©2017 Catalyzing Computing and Cybersecurity in Community Colleges (C5). This material is based upon work supported by the **National Science Foundation** under Grant No.1548315.

This is important because secret messages, however unbreakable, attract attention. For example, the Zimmerman Telegraph (1917) which directly caused the US to enter WWI was brought to the attention of US authorities long before they were able to decipher the message, simply because the message was encrypted. In some countries (e.g., China, Russia, Cuba, South Africa, etc.), in fact, it is illegal for two unlicensed counterparties to send encrypted messages. Steganography, by its very nature, attempts to conceal encrypted communications by hiding those communications in a format which is not likely to arouse suspicion (e.g., a simple JPEG image, an MP3 music file, etc.).

Detection of steganography is of considerable interest to US national cybersecurity efforts. Steganography interdiction efforts as part of our larger cybersecurity framework is referred to as Content Threat Removal. Sophisticated, military-grade software packages (e.g., Deep Secure, etc.) analyze digital content (i.e., jpg, mp3, mp4, png, wav, etc.) for hidden, secret content. The fear is that entities such as terrorist organizations could coordinate their efforts based on apparently harmless information (e.g., a photograph of animals, music files, videos, etc.) disseminated publicly such as on terrorist-controlled websites. Such digital content, for example, could appear innocuous, but, in fact, could have been manipulated in such a way that counterparties could decode, for example, coordinated attack orders.

Prerequisite Knowledge: This CCL is optimally sequenced after student work on programming is complete. Specifically, students taking this CCL should be familiar with the following programming fundamentals: conditional statements, iteration, arrays, array processing, strings, modulus mathematics, and string processing. Given its focus on programming concepts, this four-day CCL works best as a prelude to the two-day Caesar Cipher CCL, the two-day Vigenère Cipher CCL, or the two-day Symmetric and Public Key Exchange Methodologies CCL.

Length of Completion: The CCL is designed to take approximately 200-250 minutes. A programming assignment is required to be completed outside of class.

Homework: Prior to the first class of the CCL, students will read the Wikipedia article on Steganography ~~(located in Reading Folder Steganography_Wikipedia.pdf)~~ and watch the YouTube video on Steganography

Prior to the second class of the CCL, students will read the Atlantic Magazine article on the capture of Reality Winner ~~(located in Reading Folder Steganography_SecretPrinterCodes_RealityWinner.pdf)~~ and the Wikipedia article

This document is licensed with a Creative Commons Attribution 4.0 International License ©2017 Catalyzing Computing and Cybersecurity in Community Colleges (C5). This material is based upon work supported by the **National Science Foundation** under Grant No.1548315.

on Machine Information Code ~~(located in Reading Folder Steganography_MachineIdentificationCode_Wikipedia.pdf).~~

Students should submit a completed programming assignment (described below) on the day immediately after the end of this CCL.

Learning Setting: The CCL assumes that students are able to move around and work in small groups.

Lab Environment: While the CCL does not specify a particular programming language, it assumes that students have in-class access to the Internet and an IDE they are familiar with.

Activity/Lab Tasks: The CCL includes five separate activities: one In-Class Data Hiding Exercise, one In-Class Ethics Activity, one multiple choice summative assessment, one In-Class Steganography Decoding Activity, and one multi-day programming assignment.

02.Steganography_Presentation.pptx
03.Steganography_MultiDay_ProgrammingExercise.docx
Reading Links
https://en.wikipedia.org/wiki/Machine_Identification_Code
https://www.theatlantic.com/technology/archive/2017/06/the-mysterious-printer-code-that-could-have-led-the-fbi-to-reality-winner/529350/
https://en.wikipedia.org/wiki/Steganography
03.Programming Folder
Steganography_DecodeSecretMessage004FullCode.doc
Steganography_InsertSecretMessage003.doc
Steganography_MakeAllEvens002FullCode.doc
Steganography_OpenPhotograph001FullCode.doc
~~Steganography_MostSimple_DataFiles.zip~~

LEARNING OBJECTIVES AND AP CSP ALIGNMENT

Lesson Learning Objectives

Students will:

1. Articulate the definition of steganography.
2. Use programming to show how steganography can be used to create hidden information.

This document is licensed with a Creative Commons Attribution 4.0 International License ©2017 Catalyzing Computing and Cybersecurity in Community Colleges (C5). This material is based upon work supported by the **National Science Foundation** under Grant No.1548315.

3. Articulate their opinion about the use of steganography as a government surveillance tool.

ASSOCIATED AP CSP SUB LEARNING OBJECTIVES

AP COMPUTER SCIENCE PRINCIPLES COURSE, BIG IDEA 1: CREATIVE DEVELOPMENT

- LO CRD-2.C Identify input(s) to a program.
  - CRD-2.C.1 Program input is data sent to a computer for processing by a program. Input can come in a variety of forms, such as tactile, audio, visual, or text.
  - CRD-2.C.4 Inputs usually affect the output produced by a program.
  - CRD-2.C.6 Input can come from a user or other programs.
- LO CRD-2.D Identify output(s) produced by a program.
  - CRD-2.D.1 Program output is any data sent from a program to a device. Program output can come in a variety of forms, such as tactile, audio, visual, or text.
  - CRD-2.D.2 Program output is usually based on a program's input or prior state (e.g., internal values).

AP COMPUTER SCIENCE PRINCIPLES COURSE, BIG IDEA 2: DATA

- LO DAT-1.D Compare data compression algorithms to determine which is best in a particular context.
  - DAT-1.D.1 Data compression can reduce the size (number of bits) of transmitted or stored data.
  - DAT-1.D.2 Fewer bits does not necessarily mean less information.
  - DAT-1.D.3 The amount of size reduction from compression depends on both the amount of redundancy in the original data representation and the compression algorithm applied.
  - DAT-1.D.4 Lossless data compression algorithms can usually reduce the number of bits stored or transmitted while guaranteeing complete reconstruction of the original data.
  - DAT-1.D.5 Lossy data compression algorithms can significantly reduce the number of bits stored or transmitted but only allow reconstruction of an approximation of the original data.
  - DAT-1.D.6 Lossy data compression algorithms can usually reduce the number of bits stored or transmitted more than lossless compression algorithms.

This document is licensed with a Creative Commons Attribution 4.0 International License ©2017 Catalyzing Computing and Cybersecurity in Community Colleges (C5). This material is based upon work supported by the **National Science Foundation** under Grant No.1548315.

    - DAT-1.D.7 In situations where quality or ability to reconstruct the original is maximally important, lossless compression algorithms are typically chosen.
    - DAT-1.D.8 In situations where minimizing data size or transmission time is maximally important, lossy compression algorithms are typically chosen.
- LO DAT-2.C Identify the challenges associated with processing data.
    - DAT-2.C.1 The ability to process data depends on the capabilities of the users and their tools.
- LO DAT-2.D Extract information from data using a program.
    - DAT-2.D.1 Programs can be used to process data to acquire information.
- LO DAT-2.E Explain how programs can be used to gain insight and knowledge from data.
    - DAT-2.E.1 Programs are used in an iterative and interactive way when processing information to allow users to gain insight and knowledge about data.
    - DAT-2.E.2 Programmers can use programs to filter and clean digital data, thereby gaining insight and knowledge.
    - DAT-2.E.4 Insight and knowledge can be obtained from translating and transforming digitally represented information.
    - DAT-2.E.5 Patterns can emerge when data are transformed using programs.

AP COMPUTER SCIENCE PRINCIPLES COURSE, BIG IDEA 3: ALGORITHMS AND PROGRAMMING

- LO AAP-1.A Represent a value with a variable.
    - AAP-1.A.1 A variable is an abstraction inside a program that can hold a value. Each variable has associated data storage that represents one value at a time, but that value can be a list or other collection that in turn contains multiple values.
    - AAP-1.A.2 Using meaningful variable names helps with the readability of program code and understanding of what values are represented by the variables.
    - AAP-1.A.3 Some programming languages provide types to represent data, which are referenced using variables. These types include numbers, Booleans, lists, and strings.
    - AAP-1.A.4 Some values are better suited to representation using one type of data rather than another.

This document is licensed with a Creative Commons Attribution 4.0 International License ©2017 Catalyzing Computing and Cybersecurity in Community Colleges (C5). This material is based upon work supported by the **National Science Foundation** under Grant No.1548315.

- LO AAP-1.C Represent a list or string using a variable.
  - AAP-1.C.1 A list is an ordered sequence of elements. For example, [value1, value2, value3, …] describes a list where value1 is the first element, value2 is the second element, value3 is the third element, and so on.
  - AAP-1.C.3 An index is a common method for referencing the elements in a list or string using natural numbers.
  - AAP-1.C.4 A string is an ordered sequence of characters.
- LO AAP-2.D Evaluate expressions that manipulate strings.
  - AAP-2.D.1 String concatenation joins together two or more strings end-to-end to make a new string.
- LO AAP-2.G Express an algorithm that uses selection without using a programming language.
  - AAP-2.G.1 Selection determines which parts of an algorithm are executed based on a condition being true or false
- LO AAP-2.H For selection:
    - Write conditional statements.
    - Determine the result of conditional statements.
  - AAP-2.H.1 Conditional statements or "if-statements" affect the sequential flow of control by executing different statements based on the value of a Boolean expression.
- LO AAP-2.K For iteration:
    - Write iteration statements.
    - Determine the result or side-effect of iteration statements.
  - AAP-2.K.1 Iteration statements change the sequential flow of control by repeating a set of statements zero or more times, until a stopping condition is met.
- LO AAP-2.M For algorithms:
    - Create algorithms.
    - Combine and modify existing algorithms.
  - AAP-2.M.1 Algorithms can be created from an idea, by combining existing algorithms, or by modifying existing algorithms.
- LO AAP-2.N For list operations:
    - Write expressions that use list indexing and list procedures.
    - Evaluate expressions that use list indexing and list procedures.
  - AAP-2.N.1 The exam reference sheet provides basic operations on lists, including: accessing an element by index, assigning a value of an element of a list to a variable, assigning a value to an element of a

This document is licensed with a Creative Commons Attribution 4.0 International License ©2017 Catalyzing Computing and Cybersecurity in Community Colleges (C5). This material is based upon work supported by the **National Science Foundation** under Grant No.1548315.

list, inserting elements at a given index, adding elements to the end of the list, removing elements, and determining the length of a list.

- LO AAP-2.O For algorithms involving elements of a list:
  - Write iteration statements to traverse a list.
  - Determine the result of an algorithm that includes list traversals.
  - AAP-2.O.1Traversing a list can be a complete traversal, where all elements in the list are accessed, or a partial traversal, where only a portion of elements are accessed.
  - AAP-2.O.2 Iteration statements can be used to traverse a list.
- LO AAP-3.A For procedure calls:
  - Write statements to call procedures.
  - Determine the result or effect of a procedure call.
  - AAP-3.A.1 A procedure is a named group of programming instructions that may have parameters and return values.
  - AAP-3.A.2 Procedures are referred to by different names, such as method or function, depending on the programming language.
  - AAP-3.A.3 Parameters are input variables of a procedure. Arguments specify the values of the parameters when a procedure is called.
  - AAP-3.A.4 A procedure call interrupts the sequential execution of statements, causing the program to execute the statements within the procedure before continuing. Once the last statement in the procedure (or a return statement) has executed, flow of control is returned to the point immediately following where the procedure was called.
- LO AAP-3.B Explain how the use of procedural abstraction manages complexity in a program.
  - AAP-3.B.1 One common type of abstraction is procedural abstraction, which provides a name for a process and allows a procedure to be used only knowing what it does, not how it does it.
  - AAP-3.B.2 Procedural abstraction allows a solution to a large problem to be based on the solution of smaller subproblems. This is accomplished by creating procedures to solve each of the subproblems.
  - AAP-3.B.5 Using parameters allows procedures to be generalized, enabling the procedures to be reused with a range of input values or arguments.
- LO AAP-3.E For generating random values:
  - Write expressions to generate possible values.
  - Evaluate expressions to determine the possible results

This document is licensed with a Creative Commons Attribution 4.0 International License ©2017 Catalyzing Computing and Cybersecurity in Community Colleges (C5). This material is based upon work supported by the **National Science Foundation** under Grant No.1548315.

- AAP-3.E.1 The exam reference sheet provides RANDOM(a, b) which generates and returns a random integer from a to b, inclusive. Each result is equally likely to occur. For example, RANDOM (1, 3) could return 1, 2, or 3.
- AAP-3.E.2 Using random number generation in a program means each execution may produce a different result.

AP Computer Science Principles Course, Big Idea 5: Impact of Computing

- LO IOC-2.C Explain how unauthorized access to computing resources is gained.
  - IOC-2.C.1 Phishing is a technique that attempts to trick a user into providing personal information. That personal information can then be used to access sensitive online resources, such as bank accounts and emails.

LESSON DETAILS

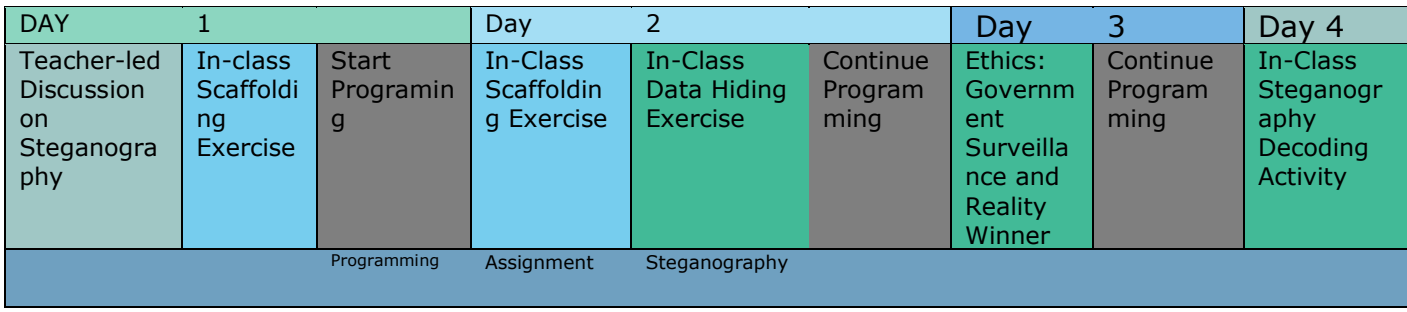
Overview of Lessons: The CCL is broken down into four 60-minute lessons..

Day 1:
Teacher-Led Discussion on Steganography (25 minutes)
In-Class Scaffolding Exercise (10 minutes)
Start Programming (25 minutes)

Day 2:
In-Class Scaffolding Exercise (20 minutes)
In-Class Data Hiding and Extraction Exercise (15 minutes)
Continue Programming (25 minutes)

Day 3:
Ethics: Government Surveillance and Reality Winner (20 minutes)
Continue Programming (40 minutes)

Day 4:
In-Class Steganography Decoding Exercise (45 minutes)



DAY 1:

This document is licensed with a Creative Commons Attribution 4.0 International License ©2017 Catalyzing Computing and Cybersecurity in Community Colleges (C5). This material is based upon work supported by the **National Science Foundation** under Grant No.1548315.

Teacher-Led Discussion (25 minutes)

02.Steganography_Presentation.pptx

The teacher cover slides 1-14 (slides 6-7 are covered on Day 3 in Ethics Discussion) of the presentation and guides a classroom discussion. To the extent possible, the teacher should reference the homework assigned for the first day of this CCL. By connecting the previous night's homework in this manner, students will feel better prepared for this new material and more comfortable raising issues and participating in this teacher-led discussion.

Slides 1-14 of the PowerPoint presentation cover a wide range of material about Steganography. Slides 6-7 are covered on Day 3 in Ethics Discussion. For teachers who may be unfamiliar with steganography, highly detailed notes are provided on presentation slides. Teachers are encouraged to familiarize themselves with the notes before instructing students with the presentation to provide students the fullest possible learning experience.

In-Class Scaffolding Exercise (10 minutes)
02.Steganography_Presentation.pptx

Slides 15-17 of the PowerPoint presentation cover the first program of the Multi-Day Programming Exercise. Teachers may wish to cover these three slides in part or in whole (as background for the student programming effort) prior to students launching into their first programming efforts. Work on this relatively simple program will prepare students for the more intellectually rigorous demands of programs two, three, and four.

Start of Programming (25 minutes)

03.Steganography_MultiDay_ProgrammingExercise.docx

Students will begin the Multi-Day Programming in class on the first day of this CCL. The teacher will look for common problems encountered by students and will encourage collaborative problem-solving without plagiarism. Teachers should supply starter code as they feel their students may require it. It is recommended that students be provided more detailed starter code for the first two programming exercises. This will help scaffold student learning and provide a solid foundation for students so that latter programming efforts can succeed. Students write a series of four short programs which advance a starting photograph file and a starting secret message file into a steganographic photograph which embeds the secret message in the photograph and then decodes it. This four-part programming exercise is scaffolded with the easiest

 This document is licensed with a Creative Commons Attribution 4.0 International License ©2017 Catalyzing Computing and Cybersecurity in Community Colleges (C5). This material is based upon work supported by the **National Science Foundation** under Grant No.1548315.

program exercise first and later programming exercises building on earlier exercises.

Students model all aspects of encrypting and decrypting a secret message in a photograph. Students will start this assignment with two photographs: a) a 400x600 pixel PNG format color photograph of their own choosing and b) a 400x600 pixel PNG format black-and-white photograph of their own choosing The presentation included with this CCL steps students through how to create these images using a Macintosh computer and free software.

DAY 2

In-Class Scaffolding Exercise (20 minutes)

Slides 18-21 of the PowerPoint presentation cover the necessary scaffolding material required for students to complete the second program of the Multi-Day Programming Exercise. Teachers may wish to cover these five slides in part or in whole (as background for the student programming effort) prior to students launching into their second programming effort. Work on this relatively simple program will prepare students for the more intellectually rigorous demands of programs three and four.

In-Class Data Hiding and Extraction Exercise: (15 minutes)
"Image Stenography Website"
This exercise should be done in groups of two.

This is an Internet-based activity that students explore the concepts of steganography in a hands-on manner. The activity consists of three parts. Firstly, students should familiarize themselves with the **Hide Image** tab of this program. Students should select a "cover" image and a "secret" image. Students should then select the number of bits which they wish to use in order to hide their "secret" image. Once done, students should save the image to their computers and share it to their partner.

Secondly, students should then use the **Unhide Image** tab to load the image encoded by their group partner. Students should then decode their group partner's stenographic image by selecting the correct number of bits.

Thirdly, students may wish to do this same encode and decode process on the photo and secret photo PNG files contained in the Steganography_OpenPhotograph001FullCode.doc located in the Programming Folder. This will be a very good warmup for the four-part programming exercise which will soon be underway.

Continue Programming (25 minutes)

This document is licensed with a Creative Commons Attribution 4.0 International License ©2017 Catalyzing Computing and Cybersecurity in Community Colleges (C5). This material is based upon work supported by the **National Science Foundation** under Grant No.1548315.

03.Steganography_MultiDay_ProgrammingExercise.docx

Slides 22-25 Students continue their efforts to complete the four programming problems. The teacher should look for common problems encountered by students and encourage collaborative problem-solving without plagiarism. Teachers should supply starter code as they feel their students may require it. It is recommended that students be provided more detailed starter code for the first two programming exercises. This will help scaffold student learning and provide a solid foundation for students so that latter programming efforts can succeed.

DAY 3

Ethics: Government Surveillance and Reality Winner (30 minutes)

Steganography_Presentation.pptx
Reading:
https://www.theatlantic.com/technology/archive/2017/06/the-mysterious-printer-code-that-could-have-led-the-fbi-to-reality-winner/529350/

Please refer to slides 6-7 of the presentation.
The teacher leads a 20-minute discussion on the privacy issues related to the case of Reality Winner. Prior to the discussion, students will have read the Atlantic Magazine article on the capture of spy Reality Winner.

Students and teacher should cover all the following issues:

1) What are the particulars of the Reality Winner case?
2) What crime was Reality Winner charged with?
3) How was Reality Winner caught?
4) What was the original purpose of Machine Identification Code Steganography as it is used in almost all common color printing devices?
5) Is the use of Steganography on unsuspecting users a violation of users' privacy as defined by the First (freedom of speech) and Fourth (unreasonable search) Amendments of the US Bill of Rights?
6) What tradeoffs must be considered with governments wish to protect the rights of its citizens while protecting those citizens from criminal activity and terrorist attack?
7) If you had been in Reality Winner's position, what would you have done?

Continue Programming (40 minutes)
03.Steganography_MultiDay_ProgrammingExercise.docx
Students continue their efforts to complete the four programming problems. The teacher should look for common problems encountered by students and

This document is licensed with a Creative Commons Attribution 4.0 International License ©2017 Catalyzing Computing and Cybersecurity in Community Colleges (C5). This material is based upon work supported by the **National Science Foundation** under Grant No.1548315.

encourage collaborative problem-solving without plagiarism. Teachers should supply starter code as they feel their students may require it. It is recommended that students be provided more detailed starter code for the first two programming exercises. This will help scaffold student learning and provide a solid foundation for students so that latter programming efforts can succeed.

DAY 4

In-Class Steganography Decoding Exercise (45 minutes)

02.Steganography_Presentation.pptx
03.Programming Folder/Steganography_DecodeSecretMessage004FullCode.doc
At this point in the CCL, students should have a) created a color, PNG format image in which a secret message has been encoded, and b) written the code which decodes a color, PNG format image containing a secret message.

In this exercise, students should share their steganographic images with their classmates (Slides 26-27). Students work individually and use their fourth program in an attempt to decode as many of the secret messages created by their classmates as time allows. Ideally, students should decode no less than three such steganographic images.

Students should share with each other how these images were created and how these images were decoded.

ACKNOWLEDGEMENTS

Resources:

Madrigal, Alexis C. "The Mysterious Printer Code That Could Have Led the FBI to Reality Winner." *The Atlantic*, 6 June 2017 (Link)

Wikipedia article on Machine Identification Code (Link)

Wikipedia article on Steganography (Link)

"Steganography [video]". alantalkstech. Oct 22, 2009.

This document is licensed with a Creative Commons Attribution 4.0 International License ©2017 Catalyzing Computing and Cybersecurity in Community Colleges (C5). This material is based upon work supported by the **National Science Foundation** under Grant No.1548315.