Whatcom COMMUNITY COLLEGE

NCYTE CENTER

VIGENÈRE CIPHER LESSON

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

VIGENÈRE CIPHER LESSON

Vigenère Ciphers fall into the category of cryptographic ciphers called One-Time Pads whose use predates Modern Cryptography by more than four centuries. This technology, while interesting, has been essentially supplanted by Modern Cryptographic methodologies (e.g., DES, RSA, etc.). Modern Cryptographic methodologies, as will be seen in the following Cybersecurity Concept Lesson (CCL), are considered *essentially* unbreakable. The *essentially* part of that sentence means that ciphers generated with Modern Cryptographic methodologies are unbreakable using current computational technologies.

While old and fully supplanted by Modern Cryptographic methodologies, perhaps, the most interesting characteristic of Vigenère Ciphers is that Vigenère Ciphers are *fully* unbreakable. The *fully* part of that sentence means that Vigenère Ciphers are not breakable using modern technologies and will not be breakable by future breakthroughs (e.g., fast methods to factor large prime numbers, quantum computing, etc.). This quality is called ***perfect-secrecy***. While it would appear that the promise of ***perfect-secrecy*** might make Vigenère Ciphers of interest to modern cryptographers, the operational overhead associated with Vigenère ciphers relegates them more to historical footnote rather than practical application.

While antiquated, Vigenère ciphers are of interest to cryptography students for three (3) reasons. Firstly, Vigenère ciphers represent an intellectual half-way point between simple Caesar Ciphers and modern key exchange methodologies. By engaging with Vigenère Ciphers at this point in their learning, high school students will be prepared and understand more deeply the mechanics of Modern Cryptographic methodologies. Secondly, Vigenère Ciphers when compared to Caesar Ciphers, present a marginal increase in computational complexity. By engaging with Vigenère Ciphers at this point in their learning, high school students will improve their programming skills by working on a problem of substance which is only marginally harder than their programming work on Caesar Ciphers. Thirdly, Vigenère Ciphers have a number of interesting qualities (e.g., ***perfect-secrecy***, burdensome operational overhead, etc.) whose understanding will help students better understand the historical context out of which Modern Cryptographic methodologies emerged.

OVERVIEW

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. For students who have not completed their study of programming a pseudocode version of the programming exercise has been included. This CCL works best as a continuation of the two-day Caesar Cipher CCL and as a lead-in to the two-day Symmetric and Public Key Exchange Methodologies

Length of Completion: The CCL is designed to take approximately 100-150 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 Vigenère Ciphers located in the Reading Folder, the YouTube video on the Vigenère Cipher, the three Khan Academy videos, and the short selection from **The Code Book** by Simon Singh. Prior to the second class of the CCL, students will read the short selection from **Blown to Bits** by Abelson, Ledeen, and Lewis, the Wikipedia article on the Venona Program (both located in the Reading Folder), and will review quickly the two websites of Venona Program-related materials provided by the US Central Intelligence Agency. In addition, students will prepare for a multiple-choice assessment which is scheduled for the end of Day 2. Students will 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 four separate activities: one In-Class Unplugged Computational Thinking exercise, one Internet Decryption activity, one multiple choice, Google Forms-based, summative assessment, and one multi-day programming assignment.

- 02.VigenèreCipher_Presentation.pptx

- 03.VigenèreCipher_InClass_ComputationalExercise.docx
- 05.VigenèreCipher_InClass_InternetDecryptionActivity.docx
- 04.VigenèreCipher_MultiDay_ProgrammingExercise.docx
- 04.Programming Folder
    - VigenèreCipher_MostSimpleFullCode_Processing.doc
    - VigenèreCipher_MostimpleFullCode_Python.doc
    - VigenèreCipher_Pseudocode.docx
- Reading Folder
    - VigenèreCipher_BlownToBitsExcerpt_HA.pdf
- Links
    - Venona Project - Wikipedia
    - Vigenère Cipher - Wikipedia

LEARNING OBJECTIVES AND AP CSP ALIGNMENT

LESSON LEARNING OBJECTIVES

Students will:

1) Articulate how a Vigenère Cipher works
2) Implement a Vigenère Cipher in a programming language of their choice
3) Articulate the role played by Vigenère Cipher during WWII and the subsequent Cold War

ASSOCIATED AP CSP SUB LEARNING OBJECTIVES

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

- LO CRD-1.A Explain how computing innovations are improved through collaboration.
    - CRD-1.A.3 Effective collaboration produces a computing innovation that reflects the diversity of talents and perspectives of those who designed it.
- LO CRD-1.C Demonstrate effective interpersonal skills during collaboration.
    - CRD-1.C.1 Effective collaborative teams practice interpersonal skills, including but not limited to: communication, consensus building, conflict resolution, and negotiation.
- LO CRD-2.B Explain how a program or code segment functions.

    - CRD-2.B.1 A program is a collection of program statements that performs a specific task when run by a computer. A program is often referred to as software.
- LO CRD-2.I For errors in an algorithm or program:
    1. Identify the error.
    2. Correct the error.
    - CRD-2.I.1 A logic error is a mistake in the algorithm or program that causes it to behave incorrectly or unexpectedly.
    - CRD-2.I.2 A syntax error is a mistake in the program where the rules of the programming language are not followed.
    - CRD-2.I.3 A run-time error is a mistake in the program that occurs during the execution of a program. Programming languages define their own run-time errors.

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.
- 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.C Evaluate expressions that use arithmetic operators.

    - AAP-2.C.1 Arithmetic operators are part of most programming languages and include addition, subtraction, multiplication, division, and modulus operators.
    - AAP-2.C.2 The exam reference sheet provides a MOD b, which evaluates to the remainder when a is divided by b. Assume that a is an integer greater than or equal to 0 and b is an integer greater than 0. For example, 17 MOD 5 evaluates to 2.
- 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:
        a. Write conditional statements.
        b. 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:
        a. Write iteration statements.
        b. 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:
        a. Create algorithms.
        b. 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:
        a. Write expressions that use list indexing and list procedures.
        b. 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

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:
    a. Write iteration statements to traverse a list.
    b. 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:
        a. Write statements to call procedures.
        b. 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:
        a. Write expressions to generate possible values.
        b. Evaluate expressions to determine the possible results

- 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 two one-hour class periods.
Day 1:

- Teacher-Led Discussion (20 minutes)
- In-Class Computational Thinking Exercise (20 minutes)
- Start of Programming (20 minutes)

Day 2:

- Continue Programming (20 minutes)
- In-Class Internet Decryption Activity (20 minutes)

DAY 1			DAY 2	
Teacher-led Discussion on Vigenère Cipher	In-Class Unplugged Computational Thinking Exercise	Start Programming	Continue Programming	In-Class Decryption Activity
	Programming	Assignment:	Vigenère	Cipher

DAY 1

Teacher-Led Discussion - 20 Minutes
02.VigenèreCipher_Presentation.pptx

The teacher should give the PowerPoint presentation and lead a classroom discussion. To the extent possible, the teacher should reference the recommended video resources on Vigenère Ciphers that students will have watched as homework for the first day of the CCL. By connecting the previous night's homework in this manner, students are better prepared for this new material and more comfortable raising issues and participating in this teacher-led discussion.

Students gain an understanding of how Vigenère Ciphers work, the historical context of these related Vigenère Ciphers, their respective advantages and disadvantages, and how Vigenère Ciphers were supplanted by more modern Cryptographic methodologies.

In-Class Unplugged Computational Thinking Exercise (20 minutes)
Students should fill out their sheets individually while working in groups of three or four. The purpose of this exercise is for students to conceptualize the steps required to program the three Vigenère Cipher algorithms: a) generation of a random Vigenère Cipher key, b) encryption of cleartext given a randomly generated Vigenère Cipher key, and c) decryption of ciphertext given the same randomly generated Vigenère Cipher key.

These algorithms challenge students programming abilities. As such, this exercise gives students a chance to discuss what steps are needed to complete the required steps for each of the three algorithms and how best to sequence those steps.

To simplify matters, the Lab Sheets provide a clear description of each algorithm, example variables, and several lines of starter code including the final return statement and the **for** loop iteration statement. Emphasizing the conceptual, creative nature of this exercise, students should draw their algorithmic steps in the shape of clouds (examples provided on each page).

At the end of the exercise, the teacher should review the actual steps of each of the three algorithms (steps listed for each algorithm on the attached Programming Exercise Description sheet). Students should mark up all the following: a) steps which they missed, b) steps which should not have been included, and c) steps which were written incorrectly.

Start of Programming (20 minutes)
Multi-day Programming Exercise: Students will write a program which takes two (2) inputs: a) a clear-text message of their choice and b) a suitable, text-

based key of their choice. The program will produce two outputs: a) a ciphertext version of the clear-text input and b) a clear-text version of the previously output ciphertext.

Students begin programming in class. The teacher will look for common problems encountered by students and encourage collaborative problem-solving without plagiarism.

DAY 2

Continue Programming (20 more minutes)
Multi-day Programming Exercise: Students will write a program which takes two (2) inputs: a) a clear-text message of their choice and b) a suitable, text-based key of their choice. The program will produce two outputs: a) a ciphertext version of the clear-text input and b) a clear-text version of the previously output ciphertext.

All of the above, with the addition that the students will have benefit of at-home work and reflection on programming problems encountered as part of their overnight programming efforts.

In-Class Unplugged Public Key Encryption Exercise (30 minutes)

This scavenger hunt activity should be done in groups of three. Students will be given a Vigenère Cipher key and single snippet of ciphertext. Using the supplied Vigenère Cipher key, students should decipher the ciphertext as quickly as possible. The cleartext generated will be a clue to the location of the next hidden ciphertext. Students should use the following website cryptography calculator for this exercise:

Vigenère Cipher Key: AYZBJEXAQZCVXVSZBB
Plaintext clues are as follows:
First Clue: DONTTHROWAWAYADUCK (next clue taped under the refuse bin)
Second Clue: QUACKBEHINDTHETIME (next clue taped behind the clock)
Last Clue: DUCKDOWNTOMYDESK (rubber duck taped under teacher's desk)
Ciphertext for the clues:
First Clue: DMMUCLOOMZYVVVVTDL (contained on lab file activity sheet)
Second Clue: QSZDTFBHYMFOEZLHNF (teacher should write this on the hidden sheet under refuse can)

Last Clue: DSBLMSTNJNOTAZKJ (teacher should write this on the hidden sheet behind the clock)

Students should write the decrypted text into the spaces provided on the Lab Exercise Sheet. The goal of this exercise is for students to find the rubber duck (or picture thereof) taped under the teacher desk

ACKNOWLEDGEMENTS

Resources:

- Wikipedia article on Vigenère Cipher (Link)
- Wikipedia article on Venona Program (Link)
- YouTube video on the Vigenère Cipher
- Khan Academy on Polyalphabetic Cipher
- Khan Academy on the One-Time Pad
- Khan Academy on Perfect-Secrecy
- **The Code Book** by Simon Singh, "Chapter 8: Le Chiffre Indéchiffrable". Pages 45-51.

PART I: THE AMERICAN RESPONSE TO SOVIET ESPIONAGE

Part I: The American Response to Soviet Espionage — Central Intelligence Agency.

PART II: SELECTED VENONA MESSAGES

Part II: Selected Venona Messages — Central Intelligence Agency.

"Chapter 5: Secret Bits How Codes Became Unbreakable". Blown to Bits: Your Life, Liberty, and Happiness After the Digital Explosion. Hal Abelson, Ken Ledeen, and Harry Lewis. Pages 169-173 (Reading Folder)