Schools Cyber Security Challenge 2 - Cryptography

https://groklearning.com/course/cyber security-challenge-2

About this activity

The world of cyber security can often be inaccessible to school students. This Challenge aims to provide students with an authentic and accessible insight into cyber security and some core concepts in cryptography.

In this Challenge, students begin with data representation and are taken on a scaffolded progression through different types of ciphers, how they can work for encryption and decryption, and some of the programming techniques necessary to implement them.

It is important to note that the key objective is to help students understand data representation and different ways to secure it before transmission, and different ways to encrypt and decrypt messages. The Challenge does not assume any prior programming experience. The programming skills that are taught are in reference to specific commands needed to encrypt and decrypt data, and as such, this Challenge is not a general programming course.

The four modules of this challenge are:

1. Representing Data
2. Cryptographic Keys
3. Frequency Analysis
4. Encrypting with binary

The learning materials in every module include notes, guided experimentation, programming activities and problems to test understanding and skills. The video resources are designed both to teach students about specific programming/cipher concepts, but also to give students a view into what working in cyber security is really like, and what people working in this field do on a day to day basis.

The people in the videos are all employees of the organisations they represent, and work in variety of roles ranging from a security analyst, all the way to the Chief Security Information Officer of a large bank.

Age

This challenge is suitable for students in all years of high school, but has been designed with the content descriptions and achievement standard of the Australian Curriculum: Digital Technologies Band 7-8 in mind. Students in later years of high school studying cryptographic techniques will still learn a lot from the course.

Language

Python — a general-purpose programming language that is widely used and easy to learn. Some of the problems don't require programming and focus instead on the cryptographic concepts.

Time

The Challenge is designed to be completed over 6-8 hours.

Key Concepts

Key Concept	**Coverage**
Abstraction	The programming solution uses a modular approach that allows specific details of the chosen cryptographic algorithms to be separated from the algorithms used to process the data. This means students can focus on smaller, more manageable aspects of the problem at any given time.
Digital systems: security	Data needs to be secured before transmission, and this is achieved through the application of different encryption algorithms. Although the algorithms used in modern cryptography are beyond the scope of this Challenge, the fundamental ideas are expressed.
Data representation	Data can be represented as strings or integers; in whole numbers or binary; or as text/images, and in variables or in collections (lists, dictionaries) stored in files.
Data interpretation	Examining the ciphertext to find patterns helps unlock the key and algorithm used to encrypt the original message.
Algorithms	Algorithms used for encryption must be reversible so that the original text can be read by the recipient. The complexity of an algorithm affects its security for cryptographic purposes.
Implementation	Programmed solutions in this Challenge cover the concepts of user input, branching, iteration, some data structures, functions and file input/output. However, many of these concepts are only addressed in brief, and would need to be explored in much greater detail to address the full expectations of the content description.

Objectives (Content Descriptions)

Digital Technologies

ACTDIK023	**Investigate how data is** transmitted and **secured** in wired, wireless and mobile networks, and how the specifications affect performance
ACTDIK024	**Investigate how digital systems represent text, image** and audio **data in binary**
ACTDIP026	**Analyse** and visualise **data using a range of software to create information**, **and use structured data to model objects** or events
ACTDIP029	Design algorithms represented diagrammatically and in English, **and trace algorithms to predict output for a given input and to identify errors**
ACTDIP030	**Implement and modify programs with user interfaces involving branching, iteration and functions in a general-purpose programming language**

What are we learning? (Abstract)

After completing the modules, students will be able to:

- Explain how text and image data are represented in digital systems
- Write and/or modify simple programs using the Python programming language
- Recognise that breaking down a problem into smaller steps (decomposition) makes it easier to solve problems
- Debug algorithms
- Recognise that steps in algorithms need to be accurate and precise
- Recognise that problems can have multiple solutions
- Utilise branching (if, if-else, and if-elif-else) in programs
- Utilise user input in programs
- Use Python-provided functions to assist with encryption and decryption algorithms
- Write functions that achieve a specific task
- Recognise that an encrypted message is only as secure as its key
- Recognise that some encryption methods are more susceptible to brute force attacks
- Appreciate that cyber security is an evolving field, with the quest to create ever more secure methods of data encryption
- Understand the career opportunities available in cyber security and related fields

Module outline

The Challenge consists of four modules:

1. Representing Data
   This module introduces the concept of data representation, and that all data is stored as numbers. Students use a range of functions (print(), ord(), chr(), input(), split()) and for loops in order to be able to represent and process characters and numbers. The module also demonstrates how messages can be stored in images (using steganography).
2. Cryptographic Keys
   This module introduces students to cryptographic keys: specifically rotation (substitution) ciphers. As some code will be run repeatedly to encrypt/decrypt, the concept of custom functions is introduced. The module goes through the process of encryption and decryption, and addresses the ease with which this cipher can be cracked using a brute force algorithm.
3. Frequency Analysis
   This module introduces students to a mixed alphabet substitution cipher as a technique that can't be broken with the previously written brute force algorithm. The technique of frequency analysis using most frequent words and letters is explored as a means to decrypt messages without a key. Students explore the Vigenère cipher as an example of a more complex (poly-alphabetic) substitution cipher.
4. Encrypting with binary

This module introduces binary representation of characters. The XOR method of encryption is introduced, where two bits of data are compared to each other. Students learn to encrypt and decrypt using XOR, exploring the security implications of having longer key lengths.

Types of component:

Discussion	**Worksheet**	**Computer-based Activity**	
Group Activity	**Unplugged Activity**	**Video**	
Animation	**Reflection**	**Game**	**App**

Challenge Introduction — why do we need to encrypt data?

People have been hiding messages for many centuries. Given so much of our lives are online, there is an increasing need to improve how systems secure our data for both storage and transmission.

One of the principles of modern organisational cyber security is to build multiple layers of defense, so that even if one layer is breached, there are subsequent layers of protection that safeguard the data. These layers of defense not only include system security (such as strong passwords and 2FA), but also physical security (controlling access to buildings, physical computing resources etc).

One such layer of system security is the encryption method (or algorithm) used on the data itself, and the security of the key (or code) used to encrypt it.

The process of translating raw data (or plaintext) into a secret code is called *encryption*. The process of translating encrypted data back into its original text is called *decryption*.

Hackers use patterns in data to force their way into systems, or find loopholes in the many layers of protection. Analysing the patterns of data to find the key (and thus break the cipher) is a commonly used method of hacking. As more complex keys are used, manually breaking ciphers can be both problematic and time-consuming. The skill of programming allows us to automate the process of applying different, prospective keys to an encrypted message. The same skills can also help us decrypt a long message with a known key.

Programming a computer means giving it instructions - a sequence of steps - to follow in a way that it “understands”. Another word for the sequence of instructions given to it is an algorithm. This Challenge will help students learn the programming skills necessary to apply some fundamental encryption and decryption techniques.

Analysing data is a skill that’s used in many industries, and increasingly used in cyber security. Data scientists working in companies analyse network traffic (for example, records of successful and unsuccessful attempts to log in to the system) to find patterns, and possibly detect signs of an impending cyber attack. Systems analysts write programs to scour through applications to test for vulnerabilities, so that they can be fixed before they are exploited by attackers to steal money/data.

The ciphers we explore in this Challenge are introductory, but these same principles are used in the cyber security industry and by hackers.

New Vocabulary

From Digital Technologies:

Algorithm: A set of rules or step by step instructions to solve a problem or achieve an objective. A recipe is an example of an algorithm - it sets out what you need and the steps you follow to combine everything to create your food item(s).

Decomposition: breaking down a problem into smaller parts which can then be dealt with individually. This allows very complicated problems to be solved by first solving their individual parts separately and then working out how those individual solutions can be used together.

Branching: Changing the instructions executed by the program based on a certain condition. This allows you to specify that your program should behave one way in some cases, but a different way in others. In Python, this is achieved through the use of `if-elif-else`.

Iteration: Repeating a set of instructions a number of times as part of the program. This allows your solution to scale to larger amounts of data and is important for generalising your solution to work for lots of similar problems. In this challenge, a `for` loop is used for iteration.

Binary: In computing, a means of representing all data using just two values. Often through the process of converting a numeric value into its binary equivalent using only 0s and 1s.

From cyber security:

Encoding: The process of converting data into a format required for processing with a computer. Decoding is the reverse - converting that data back into a human-readable form.

Steganography: A technique of hiding secret data within an ordinary, non-secret, file or message in order to avoid detection. The Challenge uses the example of hiding secret data inside an image.

Cryptography: A way of protecting information and communications by applying an algorithm and secret code (or key) such that only those for whom the data is intended (and who have the key) can read or process it.

XOR: A logical operation performed on two bits of data that outputs False (or 0) if they are the same and True (or 1) if they are different.

Activity 1.1 - Data representation and encoding

Preparation and timing

No prior knowledge is required for this activity. It can be done at the beginning of a lesson or the unit.

This activity would be best delivered as an introduction to Module 1 - Representing Data.

Overview

- What is data encoding?
- What is the difference between data and information?
- Why do computers use numbers to represent data?

Suggested Implementation

Encoding and decoding are important concepts to understand, both in digital technologies and cryptography. They allow characters, images and sound to be represented as numbers, which is the format needed by computers for processing. Without a common method of encoding, computers would not be able to communicate with each other.

Video
Representing characters with numbers

Watch the following video on YouTube, published by Bitmerge: https://youtu.be/zB85kTs-sEw

The video demonstrates the concept of characters represented by numbers, and introduces the term ASCII and Unicode. The idea of ASCII being a subset of Unicode, the relationship between uppercase and lower case letters, and the need for a common data encoding system are also discussed.

Some of the video references binary numbers, which will be covered briefly towards the end of Module 1, and more extensively in Module 4. The key bit of understanding students need to get from the video is that there is an underlying method of encoding that is used by all computers to convert between characters and numbers (called ASCII & Unicode).

Discussion
What might be the consequences if we didn't have a common method of encoding?

Everything we type, listen to and watch on computers are ultimately represented as numbers. Other computing devices (such as gaming consoles) work on exactly the same principle - that means the controls on joysticks etc also represent data as numbers.

What do you think might happen if every computer had its own way of encoding? Say my computer thought "A" was 97, but another computer manufacturer decided that "A" was represented by 200. Or my game controller sent an "accelerate" command as a 300, and another game controller's 300 was the equivalent of a stop. What could be the impacts or consequences of these inconsistencies?

Answers to the above might include:

- Computers may not be able to understand each other
- We could not play multiplayer games
- If I send a message to someone else and they have a phone from a different company, the message could look very different to what I sent (or wanted to send)
- People could not do online shopping, or banking or some examples of online and digital interactions

Discussion

What are some other conventions we use around the world to communicate?

After watching the video in Module 1 (Watch: the need for conventions), the following discussion question can be posed.

The video used the example of stop signs around the world to show how conventions determine how we communicate and understand each other. What might be some other examples of conventions used in communication?

Answers to the above might include verbal or non-verbal signs:

- A wave with the hand to signal a greeting (hello/goodbye)
- Facial expressions

Reflection Group Activity

We use conventions all the time, and computers need encoding conventions to ensure data can be interpreted in a consistent manner

Students might share examples of situations where conventions haven't been understood, for example:

- If they went to a country where they were unfamiliar with the language being spoken
- If someone spoke to them in a language they didn't understand
- If a sign/symbol was unfamiliar or ambiguous
- If a text message came through with upside down question marks or some other indication that some of the message was lost

Data encoding is a fundamental concept in cryptography. By understanding *how* data is represented, we can begin to imagine how we might be able to *manipulate* this encoding! This can be the teaser for the next lesson/activity.

Activity 1.2 - Data in colours

Preparation and timing

No prior knowledge is required for this activity.

This activity would be best delivered prior to, or in conjunction with the *Encoding messages in images* section.

Overview

- How do computers store colour data?

Suggested Implementation

Modern computing devices represent millions of colours, and a common way to represent colours is using RGB channels. Each channel has a value ranging from 0 to 255, and combined with the values from the other channels to result in a display colour.

Although there are other forms of encoding colour data (such as CMYK), this Challenge primarily looks at RGB channels.

Animation

The following link can be used to show students how the values in each of the 3 channels can be set, and when combined, form a display colour.
http://web.stanford.edu/class/cs101/image-rgb-explorer.html

Suggested activity:

- How can shades of grey be made with RGB?
- Develop a colour palette of 4/6 colours for a social media header/blog/instagram page. Students are to create a table with the colours in one column, and the corresponding RGB values in another column.
- What might be some general observations about channel values and the colour displayed?
    - Answers might include:
        - The higher the number, the lighter the colour displayed in the channel;
        - The lower the number, the darker the colour displayed in the channel;
        - When the values for r,b,g are equal, the end colour is a neutral colour;

Computer-based Activity

Students may be interested in further exploring the idea of separating and combining colour channels. This can be achieved using photo editing software such as CorelDraw, Photoshop, or the PineTools website: https://pinetools.com/rgb-channels-image

Students may open/upload an image of their choice (suitable images can be found on Pixabay). On the free PineTools website, upload the image and check/uncheck the colour channels and select ‘Decompose’.

To split the image into separate channels in Photoshop, the instructions can be found here.

To split the image into separate channels in CorelDraw, the instructions can be found here.

Discussion

How many colours can be represented in total if each colour has a range from 0 to 255?

This discussion is designed to get students thinking about the range of values available, and the combinations of the 3 channels.

Students may initially have difficulty with the concept, so a suggested scaffold would be to get them to imagine they had 2 channels, each with only 3 possible values (instead of 255). The following table provides a way to imagine this hypothetical scenario:

CHANNEL 1	CHANNEL 2
A1	B1
A2	B2
A3	B3

Working this out to find all the possibilities gives us the following values:

A1 B1	A1 B2	A1 B3
A2 B1	A2 B2	A2 B3
A3 B1	A3 B2	A3 B3

As the total number of possibilities is 9, it can be deduced that:
*total colours = number of colours in channel 1 * number of colours in channel 2*

Expanding this rule to 3 channels with 256 colours each (0-255 gives us 256 colours) results in:
Total colours = 256 * 256 * 256
= 16777216 (ie over 16 million colours!)

Reflection

RGB encoding allows us to use numbers to represent colours. This also allows us to communicate colour values easily to each other.

Students might share examples of situations where colour values need to be communicated, for example:

- Between a UX designer and a programmer
- Between a client who wants a certain colour and the web developer / graphic designer
- Between teams to ensure consistency in the work produced

Activity 2.1 - The Caesar Cipher

Preparation and timing

It is assumed that students have completed Module 1 (Data Representation). This activity would be best delivered prior to, or in conjunction with Module 2 (Rotation Ciphers).

Overview

- How does the Caesar Cipher work?

Unplugged Activity

The following resource developed the Australian Computing Academy explores the ideas in the Caesar Cipher using a series of unplugged activities.

https://cmp.ac/cipher-wheels

Computer-based Activity

The following interactive resource within the Challenge can be used by students to see how the Caesar Cipher works with different shift values:

https://groklearning.com/learn/cyber-78-py-crypto/crypto/3/

Some possible messages that students can try to encrypt with different shift values:

- their names
- their favourite food
- a sentence from their favourite book/movie

Discussion

How might some of the programming concepts that we learnt in Module 1 be useful to create a Caesar Cipher in code?

Answers to the above might include:

- We can use input() to get the value we want to shift by (the key), and the message we want to encrypt
- ord() and chr() can be used to get the underlying number representation for each letter in the original message
- print() can be used to display the encrypted message after it has been shifted

Discussion

What are the benefits of creating a Caesar Cipher through code?

Answers to the above might include:

- Creating a cipher through code means that the shift can be applied to very long messages much faster than it can be done manually.
- Using a paper cipher wheel may not result in accurate encryption, as the inner/outer wheel may move slightly or may not be perfectly aligned. This might mean that the shift is not *consistently* applied throughout the message, making it harder/impossible to decrypt.

Activity 3.1 - Frequency analysis

Preparation and timing

It is assumed that students have completed Module 1 (Data Representation) and Module 2 (Rotation Ciphers). The following unplugged activities have been designed in a manner that allows for differentiation in the classroom.

This concept is explored as the basis for Challenge Module 3 (Frequency Analysis).

Overview

- How does frequency analysis work?
- How can we use this technique to decrypt text encrypted with a mixed substitution cipher?

Unplugged Activity 1

The following website contains the names of the various monarchs of England and Britain over the last 1200 years.

Using the following table (either manually or using word processing/spreadsheet software), students are to keep tally of the number of times a monarch shares the same *first* name (e.g. Edward I and Edward II would count as the same name).

Name of Queen/King	*Tally (how many times is the same name on the list)*

Once they have finished going through the list, students share the five most popular names amongst the Kings and Queens of England and Britain.

The answers are:
- *Henry (8)*
- *Edward (8)*
- *George (6)*
- *William (2)*
- *Elizabeth (2)*

The number of times a certain monarch's name appears on the list is the *frequency*. A similar technique is used in letter frequency analysis, by analysing very large volumes of texts and keeping a tally, and converting this into percentages for each letter.

Unplugged Activity 2

The table provided in the Challenge (https://groklearning.com/learn/cyber-78-py-crypto/frequency/9/) show the most common letters in the English language.

The frequency of letters in any language is utterly dependent on the syntax and grammatical structure of that language.

Students who learn another language or speak a language other than English can explore what the letter frequencies are in their chosen language, and compare the vowels/consonant frequencies in that language to English.

https://en.wikipedia.org/wiki/Letter_frequency#Relative_frequencies_of_letters_in_other_languages

Discussion

The greater the volume of text to decrypt, the more useful letter frequency analysis is. True or False?

The most frequent letters (ETAOINSHRD etc) have been detected by analysing thousands of pieces of published texts.

If a smaller piece of text is analysed against frequency analysis e.g. the single word 'hello', the letter frequencies in this word will not correspond to the most frequent letters ETAOIN. Therefore, 'hello' encrypted using mixed alphabet substitution that gives us 'pzbbi' cannot be easily decrypted using most frequent letters.

In the discussion, students may attempt to articulate the relationship between the volume of encrypted text and the usefulness of frequency analysis. An interesting way to explore this further is to use the interactive frequency analysis tool to type/paste in increasingly larger volumes of text (say from a textbook or novel), and observe that the larger the amount of text, the closer the letter frequencies resemble ETAOIN.

Activity 3.2 - The Vigenere cipher

The Vigenere cipher is an example of a polyalphabetic substitution, but instead of a simple substitution or shift, it uses a word as a key.

Preparation and timing

It is assumed that students have completed Module 1 (Data Representation), Module 2 (Rotation Ciphers) and have understood frequency analysis and polyalphabetic substitution in Module 3 (Frequency Analysis).

This cipher is explored in the Challenge section Frequency analysis can also be beaten.

Overview

- How does the Vigenere cipher work?

Unplugged Activity

The following table shows each of the 26 letters of the alphabet progressively shifted by 1. This is commonly known as the Vigenere tableau or table, and can be used to help students understand the cipher to encrypt the plaintext “CIPHER” with the key “TURING”.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Plaintext	C	I	P	H	E	R
Key	T	U	R	I	N	G
Result						

*To find the encrypted letter, highlight the **column** that features the letter of the plaintext as the first letter, and highlight the **row** that features the letter of the key as the first letter. Where they **intersect** is the result of the Vigenere cipher ie the encrypted letter.*

The first letter of the plaintext (C) and the key (T) are looked up as follows. They intersect at 'V' which is the result of the encryption.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Plaintext	C	I	P	H	E	R
Key	T	U	R	I	N	G
Result	***V***					

Students can use this method to encrypt the remaining letters of the plaintext with the key.

Plaintext	C	I	P	H	E	R
Key	T	U	R	I	N	G
Result	***V***	***C***	***G***	***P***	***R***	***X***

Students can try encrypting a few other pieces of plaintext with a key (either set by the teacher or of the student's choosing). It is important to emphasise that the word length of the plaintext and the key should be the same.

Discussion

What might be some techniques that could be helpful to break the Vigenere cipher?

Some answers might include:

- Looking for patterns again: if the plaintext and the key are the same length, we can look for similar looking letter patterns in the encrypted text that can help us discover most common words
- If the keyword contains 3 letters, then every plaintext letter can be encrypted in 3 different ways
- Repetitions in the ciphertext can be identified, which could identify repetitions in the plaintext

Activity 4.1 - Binary numbers

Understanding how binary numbers work and their role in computers is one of the outcomes of the Digital Technologies curriculum in 7-8.

Preparation and timing

This activity can be completed at any time, no prior knowledge is needed except some numeracy and counting skills.

Binary representation is used as the basis of XOR comparisons in Challenge Module 4 (Encrypting with binary)

Overview

- What are binary numbers?
- How do they work?
- What is the relationship between binary numbers and the number 97 used to represent 'a'?

Unplugged Activity 1

The following unplugged activity can be completed as an individual or group activity. It enables students to see the relationship between decimal and binary numbers.

https://classic.csunplugged.org/wp-content/uploads/2014/12/unplugged-01-binary_numbers.pdf

Extension activity:

- What is the maximum integer value that can be represented in 3 bits? 4 bits? 7 bits? 8 bits?
- What is the significance of the 255 (as in the max value of a channel in an image), in terms of its relationship of the number of bits?

Unplugged Activity 2

Students are to represent their dates of birth as binary numbers. This requires them to understand how to convert between decimal and binary numbers.

Extension activity:

- Students are to explore the rule/pattern/algorithm to convert a decimal number into a binary number. To test their understanding, they can convert a larger number (such as their student number, population of their school/town etc) into a binary number

Game

The following game is designed to test if students can convert between decimal and binary. It allows for differentiation in the classroom, or is suitable for more older students who may already be familiar with binary numbers.

https://studio.code.org/projects/applab/iukLbcDnzqgoxuu810unLw

Many of the slides and problems include "Teacher Notes" that verified teachers can access. These provide additional information, suggestions and activities for teaching each concept and exploring ideas further with students in class.