Lesson 8: Encryption with Keys and Passwords

Adapted from code.org curriculum

Objectives: You will be able too…

- Explain the relationship between cryptographic keys and passwords
- Explain in broad terms what makes a key difficult to "crack"
- Reason about strong vs. weak passwords using a tool that shows password strength
- Understand that exponential growth is related to an encryption algorithm's strength
- Explain how and why the Vigenere cipher is a stronger form of encryption than plain substitution
- Explain properties that make for a good key when using the Vigenere Cipher

Getting Started:

- In the previous lesson you saw how relatively easy it was to crack a substitution cipher with a computational tool
- Today we'll try to crack a different code to see what it's like. Beforehand, however, we should consider why someone might want to crack a cipher in the first place

Think – Pair - Share

- Are there ethical reasons to try to crack secret codes?

Think – Pair - Share

- Are there ethical reasons to try to crack secret codes?
- People in the field of counterterrorism make a living by trying to crack the codes of other nations. Many attribute the success of the Allies in WWII to our ability to crack the Enigma code and uncover the plans of the Germans
- Others may try to crack more abstract codes that are not written by humans, searching for patterns with DNA models in order to understand their nature and be able to describe the nature of humanity

Think – Pair - Share

- It's useful to try to crack your own codes to see how strong they really are
- There are many other reasons related to mathematical exploration, pattern recognition, etc.

Encryption: Algorithms v. Keys

- Today we will attempt to crack codes, paying particular attention to the processes and algorithms that we use to do so.
- So, before starting today we want to make sure that we distinguish between an encryption algorithm and an encryption key
- **Encryption algorithm** – some method of doing encryption
- **Encryption key** – a specific input that dictates how to apply the method and can also be used to decrypt the message. Some people might say "What is the key to unlocking this message?"

Encryption: Algorithms v. Keys

- For example:
  - The Caesar Cipher is an encryption algorithm that involves shifting the alphabet
  - The amount of alphabetic shift used to encode the message is the key
  - When you are cracking the Caesar Cipher you are trying to figure out how much the alphabet was shifted - you are trying to discover the key.

Common Misconception:

- "cracking" and "decrypting" are not interchangeable
- **Decrypting** – just using an algorithm to undo the encryption. It's like using a key to unlock a lock. It's what the sender is expecting the intended recipient to do to recover the original message
- **Cracking** - more like detective work - it's like trying to pick a lock - using various methods to try to figure out what the secret message is without having or knowing the decryption "key" ahead of time.

Before we move on…

- "If random substitution is an algorithm for encryption, what is the key to a random substitution cipher?"

Before we move on…

- "If random substitution is an algorithm for encryption, what is the key to a random substitution cipher?"
  - Answer: the key is the actual letter-to-letter mapping that was used to encode the message – it can also be used to decrypt

Today…

- So, there is a difference between the algorithm (how to execute the encryption and decryption) and key (the secret piece of information)
  - In encryption you should always assume that your 'enemy' knows the encryption algorithm and has access to the same tools that you do.
  - What makes encryption REALLY strong is making it hard to guess or crack the "key," even if the "enemy" knows the encryption technique you're using.
- Today, we'll learn a little more about it and about keys and their relationship to passwords you use everyday

Another Note:

- Perhaps counter-intuitively, publicly known encryption algorithms are often more secure, since they have been exposed to a much more rigorous review by the computer science community.
- Making an encryption algorithm public allows computer scientists to verify the security of the technique either through mathematical proof, or by trying to crack it themselves.

Activity: Explore the Vigenère Cipher Widget

- We will be in Code Studio (The Vigenere Cipher – Widget)
- I'll pass out the "Exploring the Vigenere Cipher Widget – Worksheet"

Key Take-Aways:

- A well-chosen key makes a difference - there are certain keys that don't produce good results.
- We're approaching much stronger encryption because we don't need to keep the encryption method a secret.
- For example, if I told my enemy that I encrypted a message with the Vigenère cipher, my enemy would still have to do a virtually impossible amount of work to crack the code.
- Even if I told my enemy the length of the key I used, as long as that length is sufficiently large, it would still leave my enemy basically randomly guessing the key. (Even for this simplified tool, if the key is 10 letters, then there are 26^10 possible keys, ~141 trillion)

Recap: Properties of Strong Encryption

- From what you've seen what are the properties of the Vigenere Cipher that make it harder to crack? In other words, if you had to crack a vigenere cipher what would you do?

Recap: Properties of Strong Encryption

- Vigenere is strong because looking at the cipher text there are no discernable patterns assuming a good key was chosen.
- Because the ciphertext is resistant to analysis it leaves us simply having to guess what the key is.
- Even if we know the length of the key we might still have to try every possible letter combination which is a prohibitively large number of possibilities.

Recap: Properties of Strong Encryption

- For a long time, the Vigenère cipher was considered to be an unbreakable cipher and was used by governments to send important messages.
- But in the 1800s Vigenere was discovered to be susceptible to a modified form of frequency analysis. After that point it was considered insecure.
- Still the properties of Vigenere that we've found are desirable.

Activity 2: Computationally Hard Problems – How good is your password?

- We know that a good encryption algorithm reduces the problem of cracking it to simply guessing the key.
- We want the key to be Computationally Hard to guess - in other words, hard for a computer to guess.
- Computationally Hard typically means that arriving at the solution would take a computer a prohibitively long time - as in: centuries or eons.

Activity 2: Computationally Hard Problems – How good is your password?

- In terms of cracking encryption that means that the number of possible keys must be so large, that even a computer trying billions of possible keys per second is unlikely to arrive at the correct key in a reasonable amount of time.
- Nowadays when you use a password for a website or device, your password is used as a cryptographic key.
- So, choosing a good password is meaningful because we want the key to be hard for a computer to guess. How good is your password?…

Activity 2: Computationally Hard Problems – How good is your password?

- We will use the "Keys and Passwords – Worksheet"
- Click on the Code Studio – "How Secure is My Password? – Code Studio Page…"
- http://xkcd.com/936/

Wrap-Up:

- Before the Vigenere cipher was cracked, many governments openly used it. That is, they made no secret about the fact that they were using the Vigenere cipher - it was publicly known. In the modern day, it remains the case that most encryption techniques are publicly known.
- Prompt: Why might it actually be a good thing that encryption algorithms are freely shared, so that anyone who wishes can try to crack them?

Why are they freely shared?

- If the security of an encryption technique relies solely on the method remaining a secret, it actually may not be that secure.
- Ideally, a method will be so secure that even if you know which technique was used, it is difficult or impossible to crack the message.
- By making encryption techniques public, we open them up to being tested by anyone who wishes to ensure there are no clever ways of cracking the encryption.

Video: Encryption and Public Keys

- Video: https://www.youtube.com/watch?v=ZghMPWGXexs
- A few notes:
  - A Key is an input to an encryption algorithm. A password is basically the same thing.
  - Longer passwords increase the number of possible keys making it Computationally hard to guess what the key is.

What else?

- We've seen how keys relate to the strength of encryption, but we haven't seen the other side of it – how modern encryption algorithms actually work. Vigenère was cracked, so what are we using now? In order to do this, we need to understand what kinds of problems are "hard" for computers to solve.
- Right now, the only encryption we know uses a "symmetric key" – both sender and reciever need to know the secret key, and so they need to meet ahead of time.

What else?

- But is it possible for you and me to have a secure, private, encrypted exchange without meeting ahead of time and agreeing on a secret password.
  - The answer is “yes,” and we’ll find out how it works in the next lesson.

Something cool:

- How Note To Get Hacked: https://code.org/curriculum/csp/docs/hownottogethacked

Vocabulary:

- Computationally Hard – a “hard” problem for a computer is one in which it cannot arrive at a solution in a reasonable amount of time

Notes:

- Strong encryption techniques are typically publicly known algorithms, but have mathematical properties which ensure that the original message cannot easily be retrieved