



Minds in Motion

Cracking the Code to Encryption

Activity Rundown:

Have you ever wanted to send secret messages to someone without anyone being able to understand what you're saying? Whether you want to hide messages from your parents, teachers, or friends, we have a sure fire way to help you get your message across! Pay close attention to the following project and you might be able to go undetected!

You will need:

- + Paper
- + Pencil
- + Patience

Let's do it!

- 1) The first thing you will want to do is take a look at the "background" information section of this project! It is a fun read and gives a thorough layout of how encryption came to be, the different methods used, and how you can use it!
- 2) Every encryption code has 5 basic components (the first 3 are what you want to convey to the kids; the last two are pretty complicated):
 - a. Plain text (the message in normal english we want to encrypt)
 - b. Cipher text (the encrypted version of the plain text)
 - c. Key(s) (the link between the plaintext and cipher text).
 - d. Encryption Function (the mapping between the plain text and cipher text using a key).
 - e. Decryption Function (maps cipher text to plain text with a key)
- 3) The type of encryption we will be trying today is called a Caesar cipher!
- 4) For the first activity each letter will correspond with its chronological number.

A	B	C	D	E	F	G	H	I	J	K	L	M
1	2	3	4	5	6	7	8	9	10	11	12	13



Minds in Motion

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
14	15	16	17	18	19	20	21	22	23	24	25	26

- 5) Write an example text on your piece of paper to encrypt. Start with "HELLO"
- 6) Write "HELLO" on your paper and as each letter is answered write the corresponding number underneath.
 - a) You should get the answer of : 8-5-12-12-15
- 7) Next use this same sequence to build an encryption for your name.
 - a) e.g. ANDREW = 1- 14-4-18-5-23
- 8) Next come up with a key. This can be any number from 1 – 25. This key is added to the original numerical value to further develop your encryption. For example, if you decide to make your Key = 3, then you would add that to each of the values in your encryption. This can be seen in this example

Original: ANDREW
 Numbers : 1-14-4-18-5-23
 Now with the added Key=3

A	B	C	D	E	F	G	H	I	J	K	L	M
1+3=4	2+3=5	3+3=6	7	8	9	10	11	12	13	14	15	16
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
17	18	19	20	21	22	23	24	25	26	24+3=27?	28?	29?

ANDREW = 4-17-7-21-8-26

D	E	F	G	H	I	J	K	L	M	N	O	P
4	5	6	7	8	9	10	11	12	13	14	15	16
Q	R	S	T	U	V	W	X	Y	Z	A	B	C
17	18	19	20	21	22	23	24	25	26	27?	28?	29?



Minds in Motion

The message should now read DQGUHZ

- 9) In order to decrypt a message, you do the opposite. Take your encrypted message, convert it to its number format, SUBTRACT the key, and convert it back to its letter format.
- 10) When you add the key you cannot get a number larger than 26 (as this corresponds to Z, the last letter in the alphabet). If this happens, start again at 1.
- a) For example if your message is “XO” (24 15) in number format) and your key is 3, you would get 25 18 once you add the key. However, you can’t go past 26, so as soon as you reach 26 every number afterwards would start from 1 again.

A	B	C	D	E	F	G	H	I	J	K	L	M
4	5	6	7	8	9	10	11	12	13	14	15	16
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
17	18	19	20	21	22	23	24	25	26	1	2	3

- 11) The same goes when decrypting a message, you can’t go below 1.
- a) For example if your encrypted message is “DT” (4 20) and your key is 6, you would get -2 14 once you subtract the key. However, you can’t go below 1, so as soon as you reach 1 every number afterwards would count down from 26. I.e. $4 - 6 = 24$ (3...2...1...26...25...24).
- 12) Take a piece of paper and write a message and encrypt it for someone in the house or around you. Tell them what the key is, so they can decode it. If they find that too easy, have them try and decode the message without knowing the key. To do this, they would have to try every number from 1 – 25 for the key until they get a message that makes sense.



Minds in Motion

Background:

In the language of mathematics:

- The cipher text is the output of the encryption. $c = E(p, k)$, where the encryption function $E()$ takes the arguments p (the plain text) and k (the key), and performs mathematical operations on them.
- The plain text is the output of the decryption. $p = D(c, k)$, where the decryption function $D()$ takes the arguments c (the cipher text) and k (the key), and performs mathematical operations on them.

Caesar Cipher:

In a caesar cipher the 5 components are:

1. Plain text (the text you wish to encrypt)
2. Cipher text (the encrypted text)
3. Key (an Integer between 0 and 25)
4. Encryption Function: adds the value of the key to the index of each letter in the plain text to find the index of the new letter in the ciphertext.
5. Decryption Function: subtracts the value of the key from the index of each character in the ciphertext to get the index of each plain text letter.

The process of reaching the end of the alphabet and looping back around to the beginning is the concept of a modulus.

- The index of the cipher or plain text is always modified with the length of the alphabet.
-

Alphabetic Cipher:

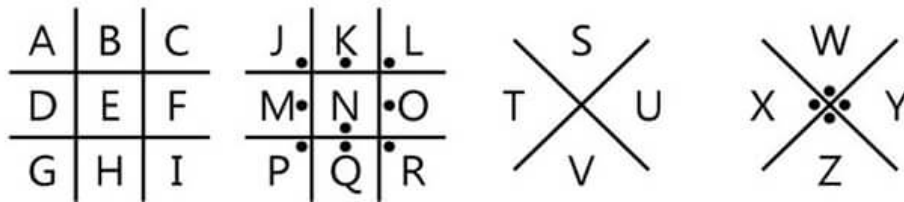
- Instead of the key being an integer, the key is 26 letters of the alphabet scrambled in any order. Encrypt by mapping the letters of the alphabet onto the key based on their index.
- Suppose your key is "XGR..." (contains all 26 letters of the alphabet)
- Then you map all A's in your message to "X"
- Map all B's in your message to "G"
- Map all C's in your message to "R", etc.
- Reverse this for decryption
- How many possible cipher texts are there for each plain text? (answer is $\sim 4 \cdot 10^{26}$, or $26!$, one for each possible key)





Minds in Motion


- What weaknesses make this scheme “crackable”?
- Cracking this code by “brute force”, i.e. by trying every possible key would take a very long time! However distribution of letters in language is not uniform, “e”, “t”, “s”, “a” are more common than “g”, “h”, “x”, “z”, etc... we can eliminate a HUGE number of possible cipher texts based on this. Moreover, repeat letters are preserved (i.e. “bottle” = “adlllef”; the “ll” shows that we have repeat letters). This again limits the possible number of cipher texts as there are some double letters that are more obvious than others.


Pig Pen Cipher:



A = 

Q = 

T = 

Z = 

FONTSEMPIRE.COM

Vernam Cipher:

- The key is an infinite list of random integers from 1 - 25 (i.e. 3 10 5 6 13 11 4 2 3 16 9 8...)
- Convert each plain text character to its integer equivalent (A =1, B =2 etc..)
- Add the each corresponding random number from the key to the index of each letter in the plain text and modulo 26 (i.e. loop back to 1 after 26, same as for the Caesar Cipher).
- For example, plain text = “HELLO” = 8 5 12 12 15
- Key = 3 10 5 6 13 11 4 2 3...

$$8 + 3 = 11$$



Minds in Motion

$$5 + 10 = 15$$

$$12 + 5 = 17$$

$$12 + 6 = 18$$

$$15 + 13 = 28 = 2$$

Cipher text = 11 15 17 18 2

- Convert this number back to a letter to make the cipher text.

i.e. cipher text = 11 15 17 18 2 = "KOQRB"

- Is this harder or easier to "crack" than the alphabetic cipher? Why?

- It is harder, it eliminates the problems we see with alphabetic ciphers (double letters, common letters)

- Is this breakable?

- In a perfect world, the only way to break this code is to try EVERY possible key. This would take an incredible amount of time!

- In practice, we cannot generate true random numbers in a computer if we can find the "random" number generator the computer used we can reduce the number of possible keys

Common problem with ciphers listed above:

They all require everyone to have a copy of everyone else's key! This is very inefficient.

RSA encryption (the best encryption scheme we have) uses a scheme where everyone has a public key and private key. You give away your public key and keep your private key. The keys are constructed so that if you encrypt something with a public key, the corresponding private key can decrypt it! RSA relies on there being no good algorithm for finding the prime factors of any VERY large number (100200 digits). HTTPS communication is based on RSA encryption.

Resources:

https://www.cerias.purdue.edu/education/k-12/teaching_resources/lessons_presentations/cryptology.html



Minds in Motion

Reach out!

We would love to hear from you about all the amazing STEM projects you are doing at home! Show us your finished products on any of the following social media platforms by tagging us or by using the following hashtags. We hope these projects have brought some excitement to your day during these difficult times.

Let us know how we did! Please [click here](#) to fill out a short survey on how well we did and what you would like to see more of in the future. Thank you!

Twitter: **@MyMindsInMotion**

Facebook: **@mindsinmotion2014 & @ucactiveliving**

Instagram: **@ucalgaryactive**

Please use the following hashtags!

#ucalgarycamps #ucalgarytogether