Funded by the Erasmus+ Programme of the European Union

Cryptography Worksheet

People have always been interested in writing secret messages. In ancient times, people had to write secret messages to keep messengers and interceptors from reading their private information. In the modern day, computers help us write secret messages to protect our credit card information, personal information, and anything sent over the internet.

How it works:

Alice and Bob want to send secret messages. They meet in private to decide what kind of key they want to use. Alice uses the secret key to write Bob messages **(encryption).** Bob uses it to figure out what Alice said **(decryption).** If Eve intercepts the message as it's being sent from Alice to Bob, we need to make sure that Eve can't figure out what they said. If she can, then we don't have a **secure cipher**.

What Makes a Good Encryption System:

1. It must be easy to encrypt. We must be able to write our message using our secret code easily. Otherwise we will spend too much time writing messages.
2. It must be easy to send. This isn't a problem with handwritten messages, but sometimes computer-generated encryptions can take a long time to send (usually because the messages become too big)
3. It must be easy to decrypt. We want to make sure our friends, family, or other recipients can easily figure out our message.
4. It must be **hard** to break. If Eve gets our message, she shouldn't be able to figure out what we said.

Caesar Shift Cipher

We write the alphabet A through Z and then shift the letters as seen below:

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 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

In this example, we shifted the letters over by one. In general, we can shift the letters over by *n* for any *n.*

Encrypting:

Alice creates a message to encode. She then shifts the alphabet over by *n.* Then for every letter in her message, she finds it in the top alphabet and then replaces it with its corresponding letter in the bottom alphabet.

Decrypting:

Bob must know the *secret key*, namely *n.* He shifts the alphabet over by *n.* He looks up each of the encoded letters in the bottom alphabet and replaces it by the corresponding letter in the top alphabet.

Problem 1:

Encrypt **THE QUICK BROWN FOX** using this Caesar Shift.

Problem 2:

Encrypt **MATH IS FUN** using a Caesar Shift of 5.

Problem 3:

Decrypt **NBSI JT GVO** using the Caesar Shift of 1 (as in our example above).

Problem 4:

Eve intercepted this message: **N QNPJ HNUMJWX.** Figure out how to break it to get Alice's message.

Problem 5:

What math are we using when encrypting and decrypting the Caesar Shift ciphers?

Problem 6:

How many different Caesar Shift ciphers are there?

Problem 7:

Alice decides the Caesar Shift cipher is too easy to break. She decides to use 50 different Caesar Shift ciphers when encrypting a message. She believes that in doing so, there are now 26^{50} different ways to choose 50 different Caesar Shift ciphers. Is Alice wrong? Why is this a good/bad idea?

Final Thought:

Should we use this cipher in modern times?

Modular Arithmetic

Perhaps you thought we didn't really use any math in the Caesar shift cipher. We can make a more "mathy" version by introducing some facts about modular arithmetic:

Modular arithmetic finds the remainder of a division problem. If we write **a (mod b)**, we are finding **r = remainder of $\frac{a}{b}$.** So our solution **r** will always be less than b.

Here are a couple examples:

5 (mod 7) = 5 since 5 < 7

4 (mod 2) = 0 since $\frac{4}{2} = 2$ with remainder 0

6 (mod 4) = 2 since $\frac{6}{4} = 1$ with remainder 2

We can add remainders together in the natural way:

If **a (mod b) = g** and **c (mod b) = h** then **(a+b) (mod b) = g+h (mod b)**. Whenever we perform an operation using modular arithmetic, we must reduce the number modulo b.

Problem 1: What is 51 (mod 4)?

Problem 2: What is (500 + 160) (mod 2)?

Problem 3: What are the only possible solutions for x (mod 4)?

Problem 4: In general, what are the possible solutions for x (mod n)?

Multiplying with modular arithmetic:

Multiplying also works in the natural way. We multiply the solutions to **x = a (mod b)** and **y = (c mod b)** and then we reduce **xy (mod b).**

Dividing is harder. Using when dividing by *n*, we multiply by the reciprocal $\frac{1}{n}$. But since we are only working with natural numbers, the fraction $\frac{1}{n}$ doesn't exist. So we need a new way to define a multiplicative inverse:

a^{-1} is the number less than b that satisfies $a^{-1}a$ (mod b) = 1

Problem 5: What is the multiplicative inverse of 5 (mod 7)?

Problem 6: What is the multiplicative inverse of 4 (mod 9)?

Problem 7: Does 2 have a multiplicative inverse (mod 4)?

Problem 8: What types of numbers *n* have multiplicative inverses for every number in the set {1, 2, …, n-1}?

Affine Ciphers

Before, when we talked about the Caesar cipher, we used the formula **m (mod 26)** where *m* stood for a letter in the alphabet. Now, we will generalize this cipher to **mx + y (mod 26)** where *m* stands for a letter in the alphabet (counting from A = 0 to Z= 25) and *x,y* are any natural number. This is called the Affine cipher.

Encrypting:

We encode as above, determining **M = mx+y (mod 26)** for each letter in the plaintext and converting this numbers to letters.

Decrypting:

To decode, Bob must know the *secret key*, namely ***x*** and ***y***. He must also know how to find a multiplicative inverse. He then can decrypt the message by using the following formula: $\mathbf{x^{-1}(M - y) \pmod{26} = m}$.

Problem 1: Encrypt **HELLO** using **x = 3, y = 1**.

Problem 2: Decrypt **MCHX** using **x = 3, y = 2.**

Problem 3: For what values of x can we use this encryption system? Think about the decryption step.

Problem 4: How many possible ciphers are there? Think about the choices for x and for y.

Final Thought: Should we use this cipher today?

Transposition Cipher

Encrypting:

Instead of shifting letters, we will now shift our entire message. We choose a number that will be the length of a row. Then we write our message in a matrix like in this example: Suppose our message is **THE SEVEN HILLS OF ROME** and our length is 5. Then we write

T H E S E
V E N H I
L L S O F
R O M E Z

and we add one z at the end to make sure we have a full matrix. Now we encode the message by reading off the columns to get **TVLRHELOENSMSHOEEIFZ**.

Decrypting:

In order for Bob to decrypt, he must know the length of Alice's matrix. This is the *secret key* of the transposition cipher. Once he knows the length of her matrix, he can create a matrix of the same size to get the message back. He would ignore the "z" since it doesn't add anything to the message.

Problem 1: Encrypt the following message using length 5: **BEWARE THE IDES OF MARCH**.

Problem 2: Decrypt the following message using length 4: **INLRREINVCFIIAOA**.

Problem 3: Eve intercepted the following message. Help her break it: **HYDAMAPOYPNZ**.

Problem 4: If Eve intercepts a message with *n* letters in it, what techniques do you use to break it?

Problem 5: How can you make your messages more secure?

Problem 6: What are the maximum number of matrices you need to try to break a message?

Final Thought: Should we use this cipher in modern day?

Bonus Problem 1: Encrypt a message using any of the ciphers we talked about today. Exchange messages with your partner, but don't tell them how you encrypted it. Try to break your partner's message.

Bonus Problem 2: Create your own cipher. Encrypt a short sentence using this cipher. Explain how it works to your partner and have him or her decipher the message.