Cryptography: Coding Lab 1

Due: Tuesday, Sep. 24, 2024

Goal: Implement a computer program utilizing modular arithmetic that can encrypt and decrypt messages using a Caesar shift cipher and a transposition cipher.

Part 1: Caesar Shift

1) Below, write pseudocode for a program that will ask for a pt message and an integer key between 1 and 26 and output the appropriate cipher text.
   Python: define a function `Cencrypt` that performs the above algorithm.

   > *Solution.*
   >
   > (a) Ask the user for a message and key.
   > (b) Ensure the key is between 1 and 26.
   > (c) Filter out non-alphabetic characters.
   > (d) Shift each letter by adding the key to the number.
   > (e) Output the cipher text.

2) How would this program need to be changed to decrypt a cipher text given a key?
   Python: define a function `Cdecrypt` and have the user choose encrypt/decrypt upon opening the program.

   > *Solution.* The difference between decryption and encryption is that encryption uses addition, and decryption uses subtraction. Or, if the algorithm is reversed, then it would be the opposite.

3) How would the decrypt function be changed to allow a brute force attack if we did not know the key?
   Python: Add an option to the decrypt function so if the person does not know the key, the program will perform a brute force attack.

   > *Solution.* We would need to iterate through each key in side the key space ($|\mathcal{K}|$). In this case, $|\mathcal{K}| = 25$, so we would need to go through 25 different iterations.

4) Check that your program works with the following message/ct pair `e(3, ‘hello world’) = KHOORZRUOG` ✓

Skip (5) because I was not in class.

6) Write the other group’s ct and your decryption of it here, along with their key.

   > *Solution.* ct = `OPNBFZ` → pt = `HIGUYS`, Key = 7

Part 2: Transposition Cipher

1) Below, write pseudocode for a program that will ask for a pt message and an integer key and output the appropriate cipher text.
Python: define a function `Tencrypt` that performs the above algorithm.

> *Solution.*
>
> (a) Ask the user for a message and the key.
>
> (b) Set an upper bound for the key (I went with the length of the ciphertext).
>
> (c) Remove any non-alphabetic characters.
>
> (d) Set the key equal to the amount of columns.
>
> (e) Calculate the rows by using integer division.
>
>   (i) If there is any remainder, we need to be sure to add 1 to the total rows so we don't leave off any of the message.
>
> (f) We need to take the string, and add it to a grid.
>
>   (i) We do this by starting at the top index
>   (ii) Work to the right until we hit the end of the column.
>   (iii) Start at the next row
>   (iv) Restart at (i) until all indices are occupied.
>
> (g) If there is any remaining space, we add a dummy character, `x`.

2) How would this program need to be changed to decrypt a cipher text given a key?
Python: define a function `Tdecrypt` and have the user choose encrypt/decrypt upon opening the program.

> *Solution.* Similar to the relationship between encryption and decryption for the Caesar cipher, we would reverse the order of assignments. We would still start at the first index, but instead of going to the right, we would go down the row.

3) How would the decrypt function be changed to allow a brute force attack if we did not know the key?
Python: Add an option to the decrypt function so if the person does not know the key, the program will perform a brute force attack.

> *Solution.* We would need to iterate through multiple grids that have varing column totals up to $|\mathcal{K}|$, where $|\mathcal{K}| = \text{len}(ct)$.

4) Check that your program works with the following message/ct pair `e(3, 'hello world') = HOLEWDLOLR` ✓

6) Write the other group's ct and your decryption of it here, along with their key.

> *Solution.* ct = `YRAOLCLK` → pt = `Ya'll Rock`, Key = 2.