

## **CSCI 357 - Cryptography**

**Ryan Kulyassa**

### **HW #3**

For my encryption function, I iterate through the plaintext, and check whether the character is a letter, as those are the only characters we want to replace. In order to find the substitute character, I check the position of the original character in the alphabet, and replace it with the letter at that same index in the key. Since the key is all caps, the original character has to be “standardized” by making it uppercase temporarily. Then, if the original character was lowercase, it is converted back to lowercase before being appended to the ciphertext string.

The decryption function follows similar logic. However, instead of first looking at the letters position in the alphabet, I first look at the position in the key. This is because we have to go “backwards” in the one-to-one mapping. Once we have the position of the character in the key, we can replace it with the letter from the original alphabet. And again we have to handle the distinction between uppercase/lowercase the same way.

Finally, the function to generate random keys works by taking a random permutation of the alphabet and returning it as a string.