**CIS-479 – LAB 1 – Buffer Overflow Attack**

**With Dr. Anys Bacha**

**Student: Demetrius Johnson**

**October 3, 2022**

# Abstract

# Part 1: Crypto Lab – Secret Key Encryption

## Task 1 – Frequency Analysis (Monoalphabetic Cipher, aka Caesar Cipher)

* I note that the tr command in Linux is the “translate command” used for translating and deleting characters. It was used to change the plaintext to all lowercased letters with no special characters except for newlines and white spaces. It was also used to encrypt (translate) the plaintext into the cipher text using the encryption key.
* These are the commands used:
  + $ tr [:upper:] [:lower:] < article.txt > lowercase.txt $ tr -cd ’[a-z][\n][:space:]’ < lowercase.txt > plaintext.txt
  + $ python >>> import random >>> s = "abcdefghijklmnopqrstuvwxyz" >>> list = random.sample(s, len(s)) >>> ’’.join(list) ’sxtrwinqbedpvgkfmalhyuojzc’
  + $ tr ’abcdefghijklmnopqrstuvwxyz’ ’sxtrwinqbedpvgkfmalhyuojzc’ \ < plaintext.txt > ciphertext.txt
* I can use the following command to change ciphertext values to their plaintext values (from frequency analysis) and make the plaintext values with capital letters so that it is easy to distinguish plaintext versus ciphertext characters, for example:
* $ tr ’aet’ ’XGE’ < in.txt > out.txt translates values ‘aet’ cipher text in in.txt to ‘XGE’ plaintext in out.txt.
* Now, I will perform frequency analysis on ciphertext.txt file (provided as one of the files I upload for this lab).
  + I used the following websites letter frequency program: <https://onlinetoolz.net/letter-frequency>

Here is the frequency analysis output from the tool

Table

Description automatically generated

Table

Description automatically generated

Table

Description automatically generated

* My strategy will be to use prepositional phrases such as “into” or “under” because they are hard to avoid in the English language and there are not very many of them. Also, I will look for the two most commonly placed together bigrams such as “it is” and also for the only single-letter English word “i”. I will also search for frequent and unavoidable bigrams and trigrams such as “it” “no” “or” “and” “to” and “too” even if the frequency is low, I know that these Boolean words are unavoidable in the English language or any language for that matter.
* After doing some frequency analysis using R programming language, I notice that the ciphertext-v appears 16 times as a **word**. I predict this value must be the letter “i” in plaintext.
  + Text

    Description automatically generated
* I continue using these strategies until I find the key.

## Task 2

# Part 2: RSA Encryption and Signature Lab

## Task 1

## Task 2

## Task 3

# Figures/diagrams:

# Conclusion

# Code (for your convenience)