Programming Spells – 1:

Python Introduction

To be able to see cryptography and security algorithms in action, you will have to rely on your code writing skills. While there are many different magical languages with which to command the great computing machine, you and your housemates will be programming in Python. It is a great language with which to tinker around with the fundamentals of security, and it is also a programming language that you will find very useful as you go out into the great big world beyond the Kingdom of Rutgers!

We are going to start with a very simple assignment: you are going to get your feet wet with programming a classic cipher in Python.

Installing Python:

To start, you need to have Python installed. If you are already familiar with Python and have it installed, then you can skip past this step. On the other hand, if you are new to Python, then I will make a recommendation on how to easily install a Python environment that will be useful for this class and, probably, future classes.

There are many ways to install Python, and perhaps the most basic is to go to python.org and download a version for whatever machine you are using. In the long run, this could lead to some extra work on your part, especially as you would need to install additional libraries for cryptography and other numerical/scientific operations.

Instead of doing that, I would recommend installing an individual copy of **Anaconda** (https://www.anaconda.com/products/individual). Anaconda is sort of an all-in-one environment for those interested in programming Python (and R) for science and engineering applications. Installing Anaconda gives you an easy way to access several different IDEs, and it comes pre-installed with most of the libraries that you will need (plus, if you have to add any, it is a piece of cake!).

Within Anaconda, you will be able to access Python through several different methods, including Spyder and Jupyter as well as from your OS's shell (e.g. cmd.exe if in Windows). I am using Spyder, though Jupyter is nice. If you are more old school, then run python from your OS shell (bonus points if you write your code with vim).

As of 2022, you have another choice, **Google Colab**, recommended by many students and Your Lovely TA. It's Jupyter running on the Google server and saving in your Google Drive associated with your Google account. (We recommend using your Rutgers account since the scarletmail is a Google service). More details are provided by TA in another document. *Hint: We recommend this option!*

Lastly, if you want to completely ignore my other suggestions, and try another "collaborative" alternative: **replit**.

Create A Program

The next part of this lesson/assignment is for you to create your own program for performing the shift cipher. In whatever development environment you choose, make your own .py file that contains the following code:

```
# -*- coding: utf-8 -*-
# Shift Cipher Implementation
# First, the alphabet used by plaintext and ciphertext

abet = 'abcdefghijklmnopqrstuvwxyz'

def enc_shift(k, plaintext):
  outciph = ''
  for ctr in plaintext.lower():
    try:
      i = (abet.index(ctr) + k)  % 26
      outciph += abet[i]
    except ValueError:
      outciph += ctr
  return outciph.lower()

plaintext = 'Alice was beginning to get very tired of sitting by her sister'
ciphertext = enc_shift(3,plaintext)
print(ciphertext)
```

Run this program. What is the resulting ciphertext output? Your house is to submit a single pdf file in Canvas containing the answers to the problems in this assignment. In the pdf file, the answer to this problem should be identified as problem 1.

Learning Some Python:

For this part of the assignment, we are going to explore some of the variables we have used.

From the Python command line, find the output of

```
>> abet.index('j')
```

Your house is to submit a single pdf file in Canvas containing the answers to the problems in this assignment. In the pdf file, the answer to this problem should be identified as problem 2.

Next, from the Python command line, find the output of

```
>> ciphertext.upper()
```

Your house is to submit a single pdf file containing the answers to the problems in this assignment. In the pdf file, the answer to this problem should be identified as problem 3.

Next, from the Python command line, find the output of

```
>> x = 'Rutgers beat'
```

```
>> x +=' Michigan State'
>> print(x)
```

Your house is to submit a single pdf file containing the answers to the problems in this assignment. In the pdf file, the answer to this problem should be identified as problem 4.

Write Some Python of Your Own:

For this part of the assignment, you are to write your own decryption function for the shift cipher. Using the code provided as an example, make a function called dec_shift() that takes as input the key k, and the ciphertext, and produces the plaintext as output. Your house is to submit a single pdf file containing the answers to the problems in this assignment. In the pdf file, the answer to this problem should be identified as problem 5.

House Internal Verification:

For this part of the assignment, each person in your House is to verify that at least two other members of your House were able to successfully complete the assignment. To do this, for example, you might want to set up a chat session between groups of three and spend 5 to 10 minutes making certain everyone could successfully write and run the code. Once a house member has verified two other people in their House, they are to add their name to a list of verifiers in problem 6. The submission of the pdf file to Canvas should only happen when all members of the group have verified two other members, and have placed their name as an answer to problem 6.

Now the big deal!!! Everyone in the House is expected to know how to do this assignment. To verify this, each House will be assigned to another House for verification. The Verifier-House will be responsible for submitting a separate verification sheet (which will be provided later). Therefore, it is super important that each House makes a concerted effort that all its members know how to do the work! And why it works!