The Use of the Historical vs. Modern Encryption

Task

1h10m - 2h

Status | Incomplete

Introduction

Do you still remember how to code in Python? In this exercise, you will use scripts to understand how encryption directly works in the code. You will see the algorithm working in real life.

Requirements

> ⚠ Note, to perform this activity successfully, you need the following:

- A Linux Virtual Machine
- A text editor
- Understanding of Python

Alberti-Vigenère Cipher Python Code

Before we try using the Vigenère cipher, let's get some further understanding of the cipher. In addition to the plain text, the Vigenère cipher also requires a keyword, which is repeated so that the total length is equal to that of the plain text. For example, suppose the plain text is `LIGHTHOUSE LABS IS THE BEST BOOTCAMP`, and the keyword is `PYTHON`.

> Then, the keyword must be repeated as follows:

Plain text: `lighthouse labs is the best bootcamp`
Keyword: `python python python python p`

Follow the tradition by removing all spaces and punctuation, converting all letters to uppercase, and dividing the result into five-letter blocks. As a result, the above plain text and keyword become the following:

Plain text: `LIGHT HOUSE LABSI STHEB ESTBO OTCAM P`
Keyword: `PYTHO NPYTH ONPYT HONPY THONP YTHON P`

After running the script, the output looks like this:

Ciphertext: `AGZOHGWMNZSGAYUZHVHRMOSGQCLAHODMMJOZE`

Exercise

Alberti-Vigenere cipher

Now it's time to experiment with Alberti-Vigenere cipher by utilizing Python.

> Follow the instructions given here to run the script given below:

- Copy and paste the Python script given below into a text editor.
- Save the text editor file.
- Open a terminal and use the following Python command to run the script: `# python3 'file_name'`
- Type the following text to encrypt: `lighthouse labs is the best bootcamp`
- Type the following keyword to use it: `python`

After running the script and entering your input, the output should read like this:

Ciphertext : `AGZOHGWMNZSGAYUZHVHRMOSGQCLAHODMMJOZE`
Original/Decrypted Text : `LIGHTTHOUSETLABSTISTTHETBESTTBOOTCAMP`

```python
#!/usr/bin/python3
#  Vigenere Cipher
# This function generates the key in a cyclic manner
# with it's length isn't equal to the length of original text
def generateKey(string, key):
    key = list(key)
    if len(string) == len(key):
        return(key)
    else:
        for i in range(len(string) -
                       len(key)):
            key.append(key[i % len(key)])
    return("" . join(key))

# This function returns the encrypted text generated
def cipherText(string, key):
    cipher_text = []
    for i in range(len(string)):
        x = (ord(string[i]) +
             ord(key[i])) % 26
        x += ord('A')
        cipher_text.append(chr(x))
    return("" . join(cipher_text))

# This function decrypts and returns the original text
def originalText(cipher_text, key):
    orig_text = []
    for i in range(len(cipher_text)):
        x = (ord(cipher_text[i]) -
             ord(key[i]) + 26) % 26
        x += ord('A')
        orig_text.append(chr(x))
    return("" . join(orig_text))

# data input
string = input("Type the Text to encrypt: ")
keyword = input("Type the keyword to use it: ")

# the input must be uppercase, so we use the function upper() with the string
key = generateKey(string.upper(), keyword.upper())
cipher_text = cipherText(string.upper(),key.upper())

# show the information.
print("Ciphertext :", cipher_text)
print("Original/Decrypted Text :",
originalText(cipher_text, key))
```

Caesar Cipher Algorithm Python Code

Like Alberti-Vigenere, Caesar cipher is another cipher you can experiment with. As you just did for Alberti-Vigenere, run the script given below using Python.

> ⚠ After running the script and entering your input, the output should read like this:

Type the Text to encrypt: `BootCamp`
Number of Shifts: `3`
Plain Text : `BootCamp`
Shift pattern : `3`
Cipher: `ErrwFdps`

```python
#!/usr/bin/python3

def encrypt(text,s):
    result = ""
  # transverse the plain text
    for i in range(len(text)):
      char = text[i]
      # Encrypt uppercase characters in plain text
      if (char.isupper()):
         result += chr((ord(char) + s-65) % 26 + 65)
      # Encrypt lowercase characters in plain text
      else:
         result += chr((ord(char) + s - 97) % 26 + 97)
    return result

string = input("Type the Text to encrypt: ")
n_shift = int(input("Number of Shifts: "))

print ("Plain Text : " + string)
print ("Shift pattern : " + str(n_shift))
print ("Cipher: " + encrypt(string,n_shift))
```

Conclusion

After this exercise, you should not only know how the Vigenère and Caesar ciphers work, but also how you can use encryption within Python. You now have a working script whenever needed to encrypt some plain text using the Caesar or Vigenère ciphers.

You may be asking yourself, “But why do I need this when I can just Google it?”. You need to understand why you are using a programming language to do this. There are multiple reasons why you would use a programming language. The first reason is because sometimes it is quicker to call or use a script to do the work for you. You may think that you can just Google it, but what if the site you typically use is down, or gets shut down forever? Then you can rely on your script to do the work for you. What if you lose an Internet connection but still need to encrypt something? Your script does not rely on having an Internet connection to run and will still work.

These are just some reasons why you would use a script to automate some of your work.

✓ Mark Completed

Previous
Cryptanalysis

Next
Cryptanalysis Quiz

W06D5

Fri Aug 2

Outline & Notes (1)

Lectures (1)

Work (10)

7 hrs

- Cryptanalysis
- The Use of the Historical vs. Modern Encryption
- Cryptanalysis Quiz
- Cryptography Features and Objectives
- Typical Levels of Cryptography
- The Use of Cryptography
- Code a Python Playfair Cipher
- Cryptographic Encryption
- Cryptographic Methods with GPG

W06D5 Schedule »