**ST. FRANCIS INSTITUTE OF TECHNOLOGY**

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**SECURITY LAB**

**Experiment – 3: Implementation of Vignere Cipher**

**Aim:** Write a program to simulate and analyze the process of Vignere Cipher.

**Objective:** After performing the experiment, the students will be able to understand the steps of vignere cipher encryption and decryption.

**Lab objective mapped:** L502.1: Students should be able to apply the knowledge of symmetric key cryptography to analyse secrecy of simple ciphers.

**Prerequisite:** Basic knowledge of cryptography.

**Requirements:** PYTHON

**Pre-Experiment Theory:**

The Vigenère cipher is a method of encrypting alphabetic text by using a simple form of polyalphabetic substitution. It is a way of encoding a message using a keyword as the key. The Vigenère cipher was developed by Giovan Battista Bellaso in the 16th century and later misattributed to Blaise de Vigenère.

Here's how the Vigenère cipher works:

1. Key Setup:

Choose a keyword that both the sender and the receiver agree upon in advance. The keyword is repeated as necessary to match the length of the plaintext message.

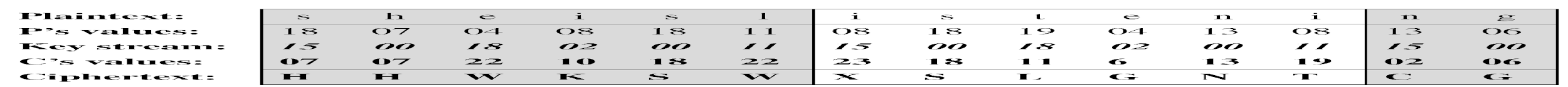
E.g. let the keyword be ‘pascal’.

Convert it into corresponding key stream. E.g. keyword ‘pascal’ = key stream ‘15 00 18 02 00 11’

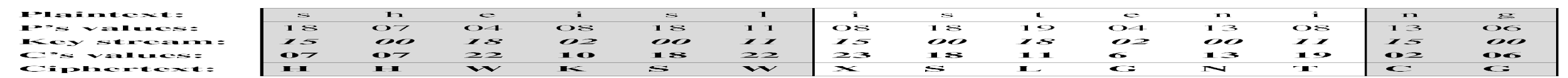
2. Encryption Process:

Let Plaintext= message = “She is Listening”

To encrypt the message, following method is used.













The encrypted message becomes "HHWKSWXSLGNTCG"

3. Decryption Process:

To decrypt any message, the receiver needs to know the Ciphertext and the keyword. Then he needs to subtract the key value from the ciphertext value. This is represented with an equation in the figure above.

The Vigenère cipher was an advancement over the more straightforward Caesar cipher since it uses a keyword, making it more challenging to break through frequency analysis. However, it is still a relatively weak cipher compared to modern encryption methods.

**Procedure:**

1. Write a program in Python to encrypt and decrypt the given input using vignere cipher.
2. Test the output of program for following Inputs:
   1. Plaintext = “She is Listening” with keyword: Pascal. Also check if decryption works.
   2. Plaintext = “The house is being Sold Tonight” with keyword: Dollars
   3. Plaintext = Your complete name (Name, middle name, surname) with keyword: Hello
3. Test the output of program for following Inputs:
   1. Ciphertext = “SMFPBZMYLWHMZYRAKPZ IS ” with keyword: HEALTH
   2. Ciphertext = “OINCMMBLSRKHJMVSJIYIITW ” with keyword: security

**Output:**

1. Attach the complete code performing encryption and decryption.
2. Attach the program output for encryption and decryption of all the inputs given above.

**Post Experimental Exercise-**

Solve the following on the journal sheets.

1. Encrypt and decrypt your complete name with keyword ‘Hello’ using vignere cipher.
2. Write in detail the strength and weaknesses of vignere cipher.

**Conclusion:**

We studied the procedure of polyalphabetic vignere cipher encryption and decryption in this experiment. The software implementation of this cipher is completed. We also explored the advantages and limitations of this cipher.

**References:** *(Add your references here)*

1. Behrouz A. Forouzan, “Cryptography & Network Security”, Tata Mc Graw Hill.
2. [www.cs.du.edu](http://www.cs.du.edu) “ENCODER/DECODER - VIGENERE CYPHER”

**LAB EXERCISE:**

1. **Write a program in Python to encrypt and decrypt the given input using vigenere cipher.**

#this function takes the input of plaintext and converts it to uppercase letters

def vigenere\_encrypt(plaintext, keyword):

def shift\_letter(letter, shift):

if letter.isalpha():

base = ord('A') if letter.isupper() else ord('a')

return chr((ord(letter) - base + shift) % 26 + base)

#base formula for encrypting the given letter

return letter

def letter\_to\_number(letter):

return ord(letter.upper()) - ord('A')

#gets the number corresponding to the letter

keyword = keyword.upper()

plaintext = plaintext.upper() #converts the letter to uppercase

encrypted\_text = [] #gives the encrypted text

keyword\_length = len(keyword)

keyword\_index = 0

for letter in plaintext: #here the alphabet is shifted according to the key

if letter.isalpha():

shift = letter\_to\_number(keyword[keyword\_index % keyword\_length])

encrypted\_text.append(shift\_letter(letter, shift))

keyword\_index += 1 #incrementing the keyword index

else:

encrypted\_text.append(letter)

return ''.join(encrypted\_text)

def vigenere\_decrypt(encrypted\_text, keyword): #function for decryption

def shift\_letter(letter, shift):

if letter.isalpha():

base = ord('A') if letter.isupper() else ord('a')

return chr((ord(letter) - base - shift) % 26 + base)

#base formula for encrypting the given letter

return letter #returns the decrypted letter

def letter\_to\_number(letter):

#function to get the the numeric value of the letter

return ord(letter.upper()) - ord('A')

keyword = keyword.upper()

encrypted\_text = encrypted\_text.upper()

decrypted\_text = []

keyword\_length = len(keyword)

keyword\_index = 0

for letter in encrypted\_text:

if letter.isalpha():

shift = letter\_to\_number(keyword[keyword\_index % keyword\_length])

decrypted\_text.append(shift\_letter(letter, shift))

keyword\_index += 1

#the value of the letter is used in the formula with the key value

else:

decrypted\_text.append(letter) #appends the letter to the decrypted text

return ''.join(decrypted\_text)

def generate\_keystream(keyword):

#Generate a keystream based on the keyword

keyword = keyword.upper()

keystream\_numbers = [str(ord(letter) - ord('A')) for letter in keyword]

return ' '.join(keystream\_numbers)

#corresponding value of every letter in the key is returned

if \_\_name\_\_ == "\_\_main\_\_":

plaintext = input("Enter your Plaintext: ");

keyword = input("Enter your key: ");

#takes the plaintext and key and returns encrypted and decrypted text.

encrypted = vigenere\_encrypt(plaintext, keyword)

decrypted = vigenere\_decrypt(encrypted, keyword)

keystream = generate\_keystream(keyword)

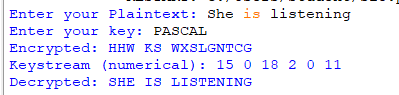
print("Encrypted:", encrypted)

print("Keystream (numerical):", keystream)

print("Decrypted:", decrypted)

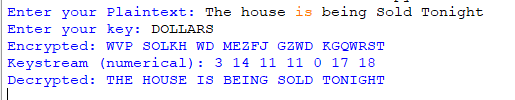
1. **Test the output of program for following Inputs:**
2. Plaintext = “She is Listening” with the keyword: Pascal. Also check if decryption works.

ENCRYPTION & DECRYPTION:



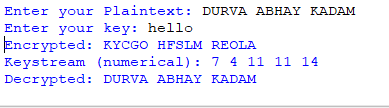
1. Plaintext = “The house is being Sold Tonight” with keyword: Dollars

ENCRYPTION & DECRYPTION:



1. Plaintext = Your complete name (Name, middle name, surname) with keyword: Hello

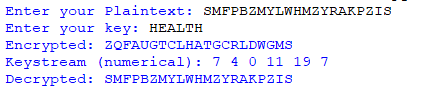
ENCRYPTION & DECRYPTION:



**3. Test the output of program for following Inputs:**

a. Ciphertext = “SMFPBZMYLWHMZYRAKPZIS ” with keyword: HEALTH

ENCRYPTION & DECRYPTION:



b. Ciphertext = “OINCMMBLSRKHJMVSJIYIITW ” with keyword: security

ENCRYPTION & DECRYPTION:

