| | |
|---|---|
| **Academic Year: 2024-25** | **Programme: BTECH-Cyber (CSE)** |
| **Year: 2nd** | **Semester: IV** |
| **Student Name: Mishitha** | **Batch : K2** |
| **Roll No: K073** | **Date of experiment: 13/1/25** |
| **Faculty: Rejo Mathew** | **Signature with Date:** |

# Experiment 2: Vigenere Cipher

**Aim:** To study and implement Vigenere Cipher.

**Learning Outcomes:**

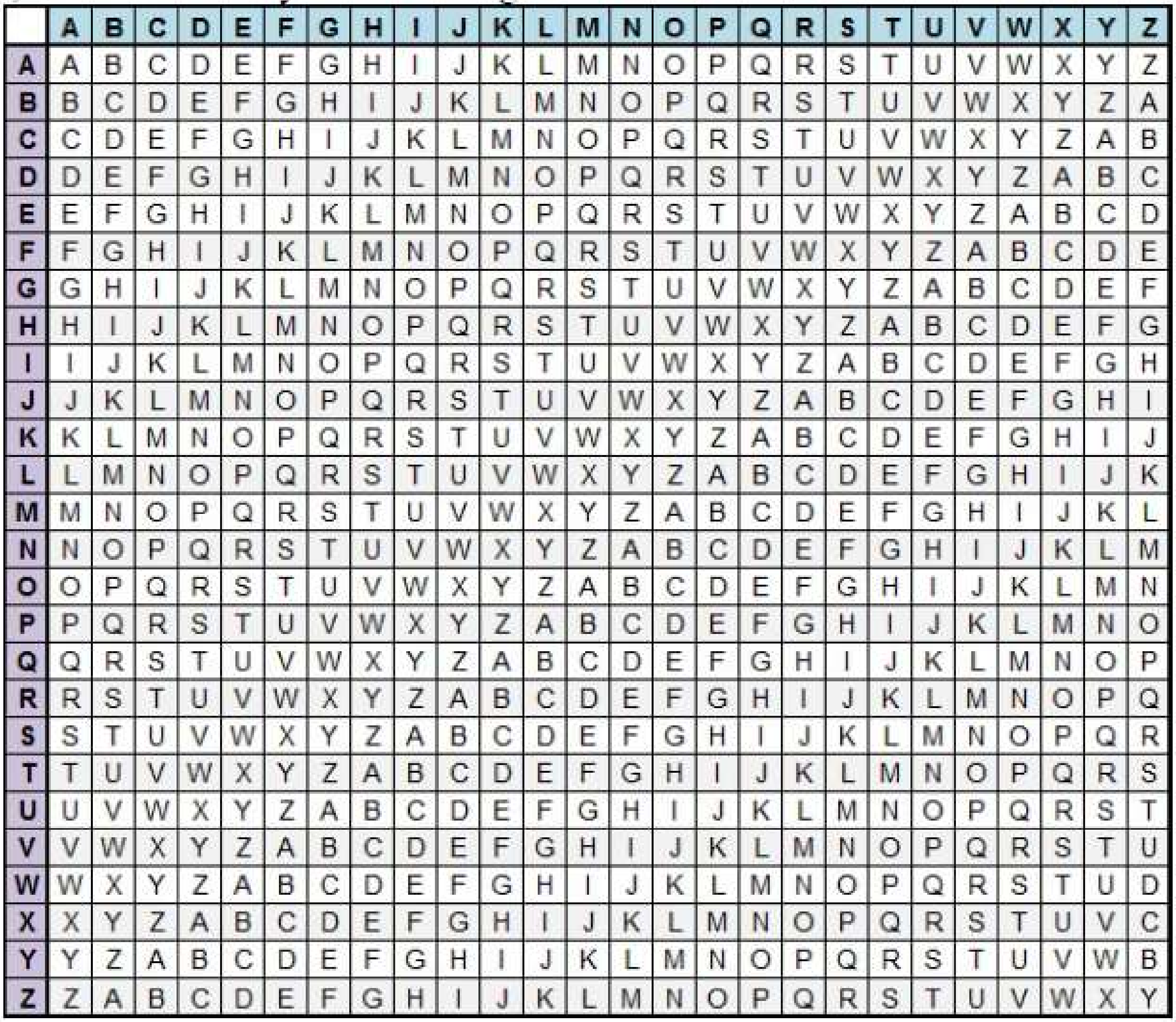
After completion of this experiment, student should be able to

1. Understand steps of Vigenere Cipher.
2. Implement Vigenere Cipher.
3. Understand variations of Vigenere Cipher and its effectiveness.

**Theory:**

The Vigenére cipher is an example of polyalphabetic substitution cipher. This cipher uses multiple one-character keys. Each of the keys encrypts one plain-text character. The first key encrypts the first plain-text character; the second key encrypts the second plain-text character, and so on. After all the keys are used, they are recycled. Thus, if we have 30 one-letter keys, every 30th character in the plain text would be replaced with the same key. This number (in this case, 30) is called the period of the cipher.
The main features of polyalphabetic substitution cipher are the following:
(a) It uses a set of related monoalphabetic substitution rules.
(b) It uses a key that determines which rule is used for which transformation.

For example, let us discuss the Vigenére cipher, which is an example of this cipher. In this algorithm, 26 Caesar ciphers make up the mono-alphabetic substitution rules. There is a shifting mechanism, from a count of 0 to 25. For each plain-text letter, we have a correspondingsubstitution, which we call the key letter. To understand this technique, we need to take a lookat a table, which is formally known as Vigenére tableau.



| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| B | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A |
| C | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B |
| D | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C |
| E | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D |
| F | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E |
| G | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F |
| H | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G |
| I | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H |
| J | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I |
| K | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J |
| L | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K |
| M | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L |
| N | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M |
| O | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N |
| P | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
| Q | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P |
| R | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q |
| S | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R |
| T | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S |
| U | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T |
| V | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U |
| W | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | D |
| X | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | C |
| Y | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | B |
| Z | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y |

The alphabet used at each point depends on a repeating keyword.

**Input:**

*Plaintext:* SEEINTHEMALL

*Keyword :* INFOSEC

**Output:**

*Ciphertext:* A R J W F X J M Z F Z D

For generating key, the given keyword is repeated in a circular manner until it matches the length of the plain text.

The keyword "INFOSEC" generates the key "INFOSECINFO"

The plain text is then encrypted using the process explained below.

### Encryption

The first letter of the plaintext, S is paired with I, the first letter of the key. So use row S and column I of the Vigenere square, namely A. Similarly, for the second letter of the plaintext, the second letter of the key is used, the letter at row E and column N is R. The rest of the plaintext is enciphered in a similar fashion.

### Decryption

Using the table, choose the row corresponding to the keyword character and look for the ciphertext character in that row.Plaintext character is then at the top of that column

**Input:** *Ciphertext:* ARJAWMPUNQZ *Keyword:* INFOSEC

**Output:** *Plaintext:* SEEINTHEMALL

### Mathematical Expression for Vigenere Cipher

Converting [A-Z] into numbers [0–25].

### Encryption

The plaintext(P) and key(K) are added modulo 26.

$E_i = (P_i + K_i) \bmod 26$

### Decryption

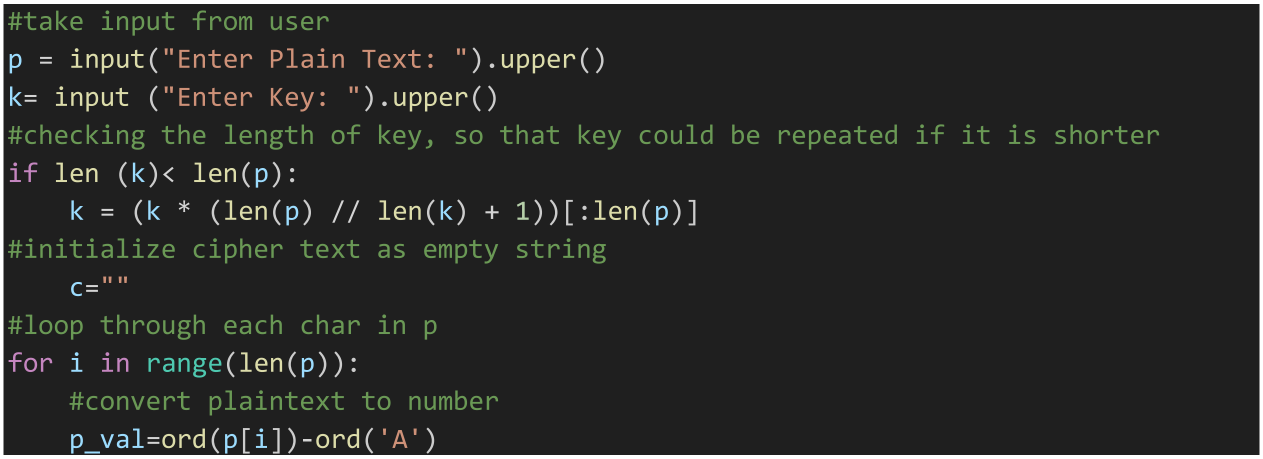
$D_i = (E_i - K_i + 26) \bmod 26$

Note: $D_i$ denotes the offset of the $i^{th}$ character of the plaintext. Like the offset of A is 0 and of B is 1 and so on.

**Steps to follow: Code has to be with comments**

**<u>Encryption</u>**

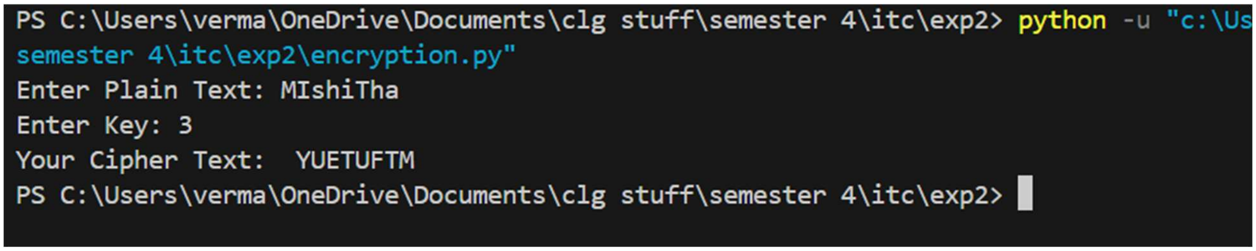
**Code:**



```
#take input from user
p = input("Enter Plain Text: ").upper()
k= input ("Enter Key: ").upper()
#checking the length of key, so that key could be repeated if it is shorter
if len (k)< len(p):
    k = (k * (len(p) // len(k) + 1))[:len(p)]
#initialize cipher text as empty string
    c=""
#loop through each char in p
for i in range(len(p)):
    #convert plaintext to number
    p_val=ord(p[i])-ord('A')
    #convert key to number
    k_val=ord(k[i])-ord('A')
    #perform encryption
    encrypval=(p_val+k_val)%26
    #converting back to character
    encrypted=chr(encrypval + ord('A'))
    #add to cipher text
    c+=encrypted

print("Your Cipher Text: ",c)
```

**Output:**



```
PS C:\Users\verma\OneDrive\Documents\clg stuff\semester 4\itc\exp2> python -u "c:\Us
semester 4\itc\exp2\encryption.py"
Enter Plain Text: MIshiTha
Enter Key: 3
Your Cipher Text:  YUETUFTM
PS C:\Users\verma\OneDrive\Documents\clg stuff\semester 4\itc\exp2>
```

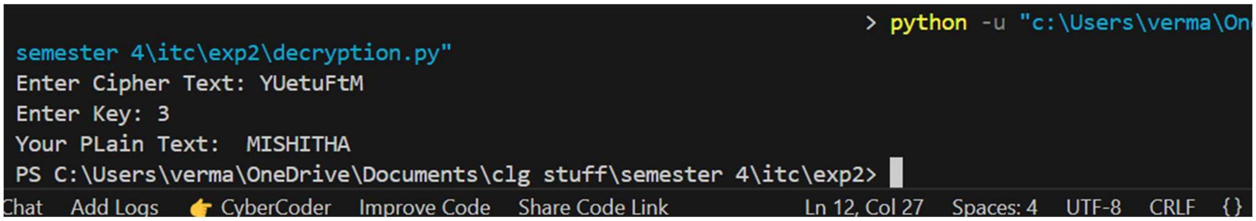
## Decryption

**Code:**

```
#take input from user
c = input("Enter Cipher Text: ").upper()
k= input ("Enter Key: ").upper()
#checking the length of key, so that key could be repeated if it is shorter
if len (k)< len(c):
    k=(k*(len(c)//len(k) + 1))[:len(c)]
#initialize plain text as empty string
    p=""
#loop through each char in cYUETUFTM
for i in range(len(c)):
    #convert ciphertext to number
    c_val=ord(c[i])-ord('A')
    #convert key to number
    k_val=ord(k[i])-ord('A')
    #perform decryption
    decrypval=(c_val-k_val+26)%26
    #converting back to character
    decrypted=chr(decrypval + ord('A'))
    #add to plain text
    p+=decrypted
```

```
print("Your PLain Text: ",p)
```

**Output:**



```
                                                      > python -u "c:\Users\verma\On
semester 4\itc\exp2\decryption.py"
Enter Cipher Text: YUetuFtM
Enter Key: 3
Your PLain Text:  MISHITHA
PS C:\Users\verma\OneDrive\Documents\clg stuff\semester 4\itc\exp2> 
```

**Questions:**

1. Is the 16th century - Vigenere Cipher still worthy? Justify?

Ans: The Vigenère Cipher, created in the 16th century, is not very strong today. While it was safe back then, modern technology can break it easily. Tools like frequency analysis and Kasiski Examination can find patterns in the cipher, especially if the key is short or used repeatedly. Today, we use stronger encryption methods like AES that are much safer and faster. So, while the Vigenère Cipher is important in history, it's not good enough to protect information now.

2. What is the common types of attacks on Vigenere Cipher?

Ans: Here are some common ways to break the Vigenère Cipher:

- Frequency Analysis: If the key is short, some letters in the ciphertext repeat more often. Attackers can use these repeating letters to guess the key.
- Kasiski Examination: This method finds repeating patterns in the ciphertext. The distance between these patterns helps attackers figure out the length of the key.
- Brute Force: This attack tries all possible keys to find the right one. If the key is short, this can be done quickly.
- Known-Plaintext Attack: If the attacker knows part of the original message, they can use that to guess the key and decrypt the rest of the message.

3. Discuss the impact of the key length on the security of the Vigenère cipher. Why does a longer key generally provide better security?

Ans: The key length affects how safe the Vigenère Cipher is. If the key is short, it repeats many times in the ciphertext, which makes it easier for attackers to find patterns.

If the key is long, it repeats less often, making it harder for attackers to spot patterns. If the key is as long as the message (called a one-time pad), the cipher is unbreakable. However, if the key is much longer than the message, it can be hard to manage.

In general, a longer key makes the cipher harder to break because it reduces repeating patterns.

4. What is difference of autokey method and keyword method of Vigenere Cipher?

Ans: There are two main ways to create the key in the Vigenère Cipher:

- Keyword Method: In this method, you pick a word (the keyword) and repeat it until it matches the length of the message. This repeated keyword is used to shift the letters in the message. For example, if the keyword is "KEY" and the message is "HELLO," the keyword would be repeated as "KEYKE," and the letters of the message would be shifted based on the keyword.

- Autokey Method: This method starts with the keyword, but after that, the plaintext itself is used to continue the key. For example, if the keyword is "KEY" and the message is "HELLO," the key starts as "KEY" and then continues with "HELLO." This makes the key longer and harder to guess, which makes the cipher more secure.

In short, the autokey method is more secure than the keyword method because it uses the message itself to help create the key, making the key harder to predict.

**Conclusion:**

In conclusion, this experiment taught us how to encrypt and decrypt messages using the Vigenère cipher. We used a key to shift the letters in the message and repeated the key if it was too short. The encryption formula added the key’s letters to the message, while the decryption formula reversed this. It was important to make sure the key was the right length and used correctly. This exercise helped us understand how encryption works and how small changes can make a big difference in keeping messages safe.