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.

Plaintext:	s	h	e	i	s	l	i	s	t	e	n	i	n	g
P’s values:	18	07	04	08	18	11	08	18	19	04	13	08	13	06
Key stream:	***15***	***00***	***18***	***02***	***00***	***11***	***15***	***00***	***18***	***02***	***00***	***11***	***15***	***00***

Encryption: $C_i = P_i + k_i$ Decryption: $P_i = C_i - k_i$

C’s values:	**07**	**07**	**22**	**10**	**18**	**22**	**23**	**18**	**11**	**6**	**13**	**19**	**02**	**06**
Ciphertext:	**H**	**H**	**W**	**K**	**S**	**W**	**X**	**S**	**L**	**G**	**N**	**T**	**C**	**G**

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:
    a. Plaintext = "She is Listening" with keyword: Pascal. Also check if decryption works.
    b. Plaintext = "The house is being Sold Tonight" with keyword: Dollars
    c. Plaintext = Your complete name (Name, middle name, surname) with keyword: Hello
3. Test the output of program for following Inputs:
    a. Ciphertext = "SMFPBZMYLWHMZYRAKPZIS " with keyword: HEALTH
    b. 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 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.

In Lab Exercise (Implementation of vignere cipher.):

```
def vigenere_encrypt(plaintext, keyword):
    # Convert keyword and plaintext to uppercase to ensure case insensitivity
    keyword = keyword.upper()
    plaintext = plaintext.upper()

    # Initialize an empty string to store the encrypted text
    cipher_text = ""

    # Start at the beginning of the keyword
    keyword_index = 0

    # Iterate over each character in the plaintext
    for char in plaintext:
        # Check if the character is a letter (ignoring non-alphabetic characters)
        if char.isalpha():
            # Calculate the shift value based on the current keyword character
            shift = ord(keyword[keyword_index]) - ord('A')

            # Encrypt the character by shifting it according to the Vigenère cipher formula
            encrypted_char = chr(((ord(char) - ord('A') + shift) % 26) + ord('A'))

            # Append the encrypted character to the cipher_text
            cipher_text += encrypted_char

            # Move to the next character in the keyword, wrapping around if necessary
            keyword_index = (keyword_index + 1) % len(keyword)
        else:
            # If the character is not a letter, just append it as is
            cipher_text += char
```

```
    # Return the fully encrypted text
    return cipher_text

def vigenere_decrypt(cipher_text, keyword):
    # Convert keyword and cipher_text to uppercase to ensure case
insensitivity
    keyword = keyword.upper()
    cipher_text = cipher_text.upper()

    # Initialize an empty string to store the decrypted text
    plaintext = ""

    # Start at the beginning of the keyword
    keyword_index = 0

    # Iterate over each character in the cipher text
    for char in cipher_text:
        # Check if the character is a letter (ignoring
non-alphabetic characters)
        if char.isalpha():
            # Calculate the shift value based on the current keyword
character
            shift = ord(keyword[keyword_index]) - ord('A')

            # Decrypt the character by shifting it backwards
according to the Vigenère cipher formula
            decrypted_char = chr(((ord(char) - ord('A') - shift +
26) % 26) + ord('A'))

            # Append the decrypted character to the plaintext
            plaintext += decrypted_char

            # Move to the next character in the keyword, wrapping
around if necessary
            keyword_index = (keyword_index + 1) % len(keyword)
        else:
            # If the character is not a letter, just append it as is
```

```
            plaintext += char

    # Return the fully decrypted text
    return plaintext

# Main program loop
while True:
    # Display menu options to the user
    choice = int(input("MENU \n1. Encrypt \n2. Decrypt \n3. Exit
\nEnter Your Choice: "))

    # Check if the user chose to encrypt text
    if choice == 1:
        text = input("Enter the text: ")
        keyword = input("Enter the keyword: ")
        # Call the encryption function and print the result
        encrypted_text = vigenere_encrypt(text, keyword)
        print(f"Encrypted Text: {encrypted_text}")

    # Check if the user chose to decrypt text
    elif choice == 2:
        text = input("Enter the text: ")
        keyword = input("Enter the keyword: ")
        # Call the decryption function and print the result
        decrypted_text = vigenere_decrypt(text, keyword)
        print(f"Decrypted Text: {decrypted_text}")

    # Check if the user chose to exit the program
    elif choice == 3:
        break

    # Handle invalid choices
    else:
        print("Invalid Choice")
```

Test the output of program for following Inputs:

a. Plaintext = "She is Listening" with keyword: Pascal. Also check if decryption works.

Encrypt:

```
MENU
1. Encrypt
2. Decrypt
3. Exit
Enter Your Choice: 1
Enter the text: She is Listening
Enter the keyword: pascal
Encrypted Text: HHW KS WXSLGNTCG
```

Decrypt:

```
MENU
1. Encrypt
2. Decrypt
3. Exit
Enter Your Choice: 2
Enter the text: HHW KS WXSLGNTCG
Enter the keyword: pascal
Decrypted Text: SHE IS LISTENING
```

b. Plaintext = "The house is being Sold Tonight" with keyword: Dollars

Encrypt

```
MENU
1. Encrypt
2. Decrypt
3. Exit
Enter Your Choice: 1
Enter the text: The house is being Sold Tonight
Enter the keyword: Dollars
Encrypted Text: WVP SOLKH WD MEZFJ GZWD KGQWRST
```

Decrypt

```
MENU
1. Encrypt
2. Decrypt
3. Exit
Enter Your Choice: 2
Enter the text: WVP SOLKH WD MEZFJ GZWD KGQWRST
Enter the keyword: Dollars
Decrypted Text: THE HOUSE IS BEING SOLD TONIGHT
```

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

Encrypt

```
MENU
1. Encrypt
2. Decrypt
3. Exit
Enter Your Choice: 1
Enter the text: Vishal Rajesh Mahajan
Enter the keyword: Hello
Encrypted Text: CMDSOS VLUSZL XLVHNLY
```

Decrypt

```
MENU
1. Encrypt
2. Decrypt
3. Exit
Enter Your Choice: 2
Enter the text: CMDSOS VLUSZL XLVHNLY
Enter the keyword: Hello
Decrypted Text: VISHAL RAJESH MAHAJAN
```

Test the output of program for following Inputs:

a. Ciphertext = "SMFPBZMYLWHMZYRAKPZIS " with keyword: HEALTH

```
MENU
1. Encrypt
2. Decrypt
3. Exit
Enter Your Choice: 2
Enter the text: SMFPBZMYLWHMZYRAKPZIS
Enter the keyword: Health
Decrypted Text: LIFEISFULLOFSURPRISES
```

b. Ciphertext = "OINCMMBLSRKHJMVSJIYIITW " with keyword: security

```
MENU
1. Encrypt
2. Decrypt
3. Exit
Enter Your Choice: 2
Enter the text: OINCMMBLSRKHJMVSJIYIITW
Enter the keyword: Security
Decrypted Text: WELIVEINANINSECUREWORLD
```