|  |  |
| --- | --- |
| **Academic Year: 2024-25** | **Programme: BTECH-Cyber (CSE)** |
| **Year: 2nd** | **Semester: IV** |
| **Student Name: Jal Bafana** | **Batch: K1** |
| **Roll No: K005** | **Date of experiment: 09.01.2025** |
| **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.

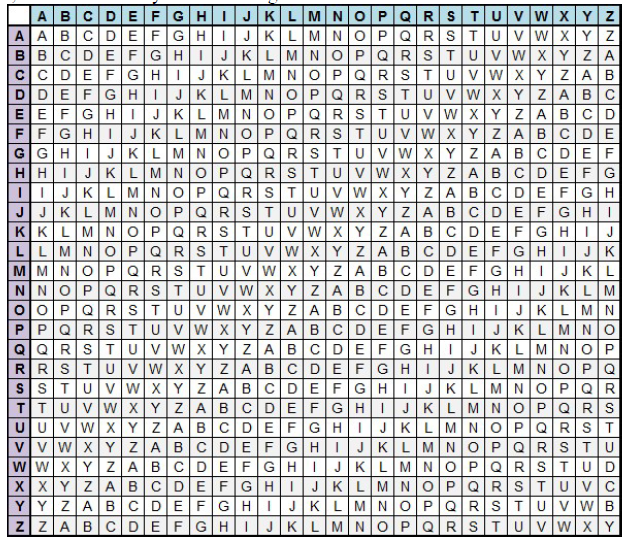
**Theory:**

The Vigenére cipher is an example of polyalphabetic substitution cipher. This cipher usesmultiple one-character keys. Each of the keys encrypts one plain-text character. The first keyencrypts 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 thiscase,30)iscalledtheperiodofthecipher.

Themainfeaturesofpolyalphabeticsubstitutioncipherarethefollowing:

1. Itusesasetofrelatedmonoalphabeticsubstitutionrules.
2. Itusesakeythatdetermineswhichruleisusedforwhichtransformation.

For example, let us discuss the Vigenére cipher, which is an example of this cipher. In thisalgorithm,26Caesarciphersmakeupthemono-alphabeticsubstitutionrules.Thereisashifting mechanism, from a count of 0 to 25. For each plain-text letter, we have a correspondingsubstitution,whichwecallthekeyletter.Tounderstandthistechnique,weneedtotakealookatatable,whichisformallyknownasVigenéretableau.



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 repeatedin 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.

Ei = (Pi + Ki) mod 26

**Decryption**

Di = (Ei - Ki + 26) mod 26

Note: Di denotes the offset of the ith character of the plaintext. Like offset of A is 0 and of B is 1 and so on.

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

**Encryption:**

#include <iostream>

using namespace std;

int main() {

char text[100], key[100], encryptedText[100];

int textLen = 0, keyLen = 0;

cout << "Enter the text: ";

cin.getline(text, 100);

cout << "Enter the key: ";

cin.getline(key, 100);

// Calculate the length of the text and key

while (text[textLen] != '\0') textLen++;

while (key[keyLen] != '\0') keyLen++;

// Encrypt the text

for (int i = 0, j = 0; i < textLen; i++) {

// Encrypt alphabetic characters

if ((text[i] >= 'A' && text[i] <= 'Z') || (text[i] >= 'a' && text[i] <= 'z')) {

// Encrypt uppercase letters

if (text[i] >= 'A' && text[i] <= 'Z') {

encryptedText[i] = char(((text[i] - 'A' + (key[j % keyLen] - 'A')) % 26) + 'A');

}

// Encrypt lowercase letters

else {

encryptedText[i] = char(((text[i] - 'a' + (key[j % keyLen] - 'a')) % 26) + 'a');

}

j++; // Move to the next character in the key

} else {

encryptedText[i] = text[i]; // Non-alphabetic characters remain unchanged

}

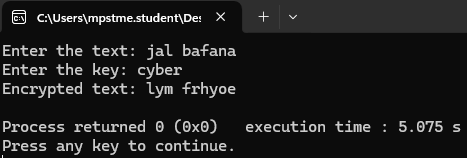
}

encryptedText[textLen] = '\0'; // Null-terminate the encrypted text

cout << "Encrypted text: " << encryptedText << endl;

return 0;

}

****

**Decryption:**

#include <iostream>

using namespace std;

int main() {

char encryptedText[100], key[100], decryptedText[100];

int textLen = 0, keyLen = 0;

cout << "Enter the encrypted text: ";

cin.getline(encryptedText, 100);

cout << "Enter the key: ";

cin.getline(key, 100);

// Calculate the length of the encrypted text and key

while (encryptedText[textLen] != '\0') textLen++;

while (key[keyLen] != '\0') keyLen++;

// Decrypt the text

for (int i = 0, j = 0; i < textLen; i++) {

// Decrypt alphabetic characters

if ((encryptedText[i] >= 'A' && encryptedText[i] <= 'Z') || (encryptedText[i] >= 'a' && encryptedText[i] <= 'z')) {

// Decrypt uppercase letters

if (encryptedText[i] >= 'A' && encryptedText[i] <= 'Z') {

decryptedText[i] = char(((encryptedText[i] - 'A' - (key[j % keyLen] - 'A') + 26) % 26) + 'A');

}

// Decrypt lowercase letters

else {

decryptedText[i] = char(((encryptedText[i] - 'a' - (key[j % keyLen] - 'a') + 26) % 26) + 'a');

}

j++; // Move to the next character in the key

} else {

decryptedText[i] = encryptedText[i]; // Non-alphabetic characters remain unchanged

}

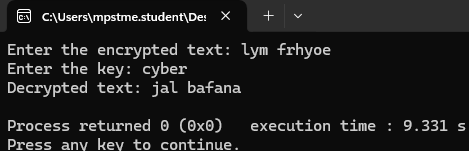
}

decryptedText[textLen] = '\0'; // Null-terminate the decrypted text

cout << "Decrypted text: " << decryptedText << endl;

return 0;

}



**Questions:**

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

The Vigenère cipher, though historically significant, is no longer considered secure for modern cryptographic needs. It relies on a repeating key and is vulnerable to frequency analysis and other modern attack techniques like the Kasiski examination. However, it holds value in educational contexts for understanding the basics of encryption and cryptanalysis, serving as a foundation for studying more advanced ciphers.

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

The most common attacks on the Vigenère cipher include:

* **Frequency Analysis**: Exploits the repetition of patterns in the ciphertext when the key is short.
* **Kasiski Examination**: Identifies the key length by analyzing repeating patterns in the ciphertext.
* **Friedman Test (Index of Coincidence)**: Estimates the key length by comparing the distribution of letters in the ciphertext.  
  These methods expose weaknesses due to the cipher's reliance on a periodic key.

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

The security of the Vigenère cipher is directly linked to the length of its key:

* **Short Key**: Repetition makes it susceptible to frequency analysis and Kasiski examination.
* **Long Key**: A key as long as the plaintext (one-time pad) ensures high security, as there are no repeating patterns.  
  A longer key reduces predictability, making cryptanalysis significantly harder. However, even with a long key, modern computers and techniques can break it given sufficient time.

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

* **Keyword Method**: The key is a short, repeating keyword. This introduces patterns in the encryption, making it vulnerable to cryptanalysis.
* **Autokey Method**: The key starts with a keyword but then incorporates the plaintext itself to extend the key. This reduces repetition and increases security, but it still has vulnerabilities if the plaintext is guessed or known.

**Conclusion:** *In conclusion, we learned about the Vigenère cipher and implemented its encryption and decryption processes through coding, gaining a practical understanding of its workings.*