

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**Lab Report**

**Course Title: Cryptography and Network Security Lab**

**Course Code: CSE–432**

|  |  |
| --- | --- |
| **Submitted By** | **Submitted To** |
| **Name:** Mst. Homai Ara Yesmin  **ID:** 221311139  **Section:** D  **Semester:** 8th  **Batch:** 29th  **Dept. Of CSE**  **Varendra University,**  **Rajshahi** | **Mohammad Faisal Al-Naser**  Lecturer  **Md. Fayzul Islam**  Lecturer  **Dept. Of CSE**  **Varendra University,**  **Rajshahi** |

**Submission date: 31.07.2025**

**Experiment No: 01**

**Experiment Name: Caesar Cipher Implementation**

**Input:**

* A string containing the plaintext.
* An integer key representing the shift amount.

**Encryption Steps:**

1. Iterate through each character in the plaintext.

2. If the character is an uppercase letter:

* Shift it forward by the key positions within the range 'A' to 'Z'.

3. If the character is a lowercase letter:

* Shift it forward by the key positions within the range 'a' to 'z'.

4. If the character is non-alphabetic:

* Leave it unchanged.

5. Concatenate the result to form the ciphertext.

**Decryption Steps:**

* Perform the same process but shift in the opposite direction by using (26 - key).

**Code:**

|  |  |
| --- | --- |
| #include <iostream>  using namespace std;  int main() {  char alpha[] = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";  string plaintext = "SHRABONY";  string ciphertext = "";  string decryptedtext = "";  int key = 3;  for (int i = 0; i < plaintext.length(); i++) {  char ch = plaintext[i];  int index = -1;  for (int j = 0; j < 26; j++) {  if (alpha[j] == ch) {  index = j;  break;  }}  int newIndex = (index + key) % 26;  ciphertext += alpha[newIndex];  } | cout << "Ciphertext (Encrypted): " << ciphertext << endl;  for (int i = 0; i < ciphertext.length(); i++) {  char ch = ciphertext[i];  int index = -1;  for (int j = 0; j < 26; j++) {  if (alpha[j] == ch) {  index = j;  break;  } }  int newIndex = (index - key + 26) % 26;  decryptedtext += alpha[newIndex];  }  cout << "Decrypted (Original Plaintext): " << decryptedtext << endl;  return 0;  } |

**Output:**

