

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**Lab Report-03**

**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:** 9th  **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: 15.08.2025**

**Experiment No: 03**

**Experiment Name: Affine Cipher for Extended ASCII (0–255)**

**Code:**

|  |  |
| --- | --- |
| #include <iostream>  #include <string>  using namespace std;  const int M = 256;  int modInverse(int a, int m) {  a = a % m;  for (int x = 1; x < m; x++)  if ((a \* x) % m == 1)  return x;  return -1;  }  string encrypt(string text, int a, int b) {  string result = "";  for (unsigned char c : text) {  int x = c;  int enc = (a \* x + b) % M;  result += static\_cast<unsigned char>(enc);  }  return result;  }  string decrypt(string text, int a, int b) {  string result = "";  int a\_inv = modInverse(a, M);  if (a\_inv == -1) return "Invalid 'a' (no modular inverse)";  for (unsigned char c : text) {  int y = c;  int dec = (a\_inv \* (y - b + M)) % M;  result += static\_cast<unsigned char>(dec);  }  return result;  } | int main() {  string plaintext;  int a, b;  cout << "Enter plaintext: ";  getline(cin, plaintext);  cout << "Enter key a (coprime with 256): ";  cin >> a;  cout << "Enter key b: ";  cin >> b;  if (modInverse(a, M) == -1) {  cout << "Error: 'a' must be coprime with 256." << endl;  return 0;  }  string cipher = encrypt(plaintext, a, b);  cout << "Encrypted: " << cipher << endl;  string decrypted = decrypt(cipher, a, b);  cout << "Decrypted: " << decrypted << endl;  return 0;  } |

**Output:**

