



বরেন্দ্র বিশ্ববিদ্যালয়

বরেন্দ্র
বিশ্ববিদ্যালয়

V A R E N D R A U N I V E R S I T Y

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Lab Report-2

Course Title: Cryptography and Network Security Lab

Course Code: CSE-432

Submitted By	Submitted To
Name: Md. Nahid Hasan ID: 221311131 Section: D Semester: 9 th Batch: 29 th Dept. Of CSE Varendra University, Rajshahi	Mohammad Faisal Al-Naser Lecturer Md. Fayzul Islam Lecturer Dept. Of CSE Varendra University, Rajshahi

Submission Date: 08.08.2025

❑ Experiment No: 02

❑ Experiment Name: Implementation of Playfair Cipher Algorithm

❖ Introduction:

The plaintext is prepared by breaking it into pairs, inserting 'X' between repeated letters, and adding an 'X' if the length is odd. Each pair is encrypted based on their positions in the matrix using three rules:

1. If both letters are in the same row, replace each with the letter to its right.
2. If both are in the same column, replace each with the letter below it.
3. If neither condition applies, replace them with letters in the same row but in the column of the other letter.

❖ Code

```
#include <iostream>
#include <string>
#include <algorithm>
using namespace std;

char matrix[5][5];

string prepareKey(string key) {
    string result = "";
    for (char &c : key) {
        if (isalpha(c)) {
            c = toupper(c);
            if (c == 'J') c = 'I'; // replace J with I
            if (result.find(c) == string::npos)
                result += c;
        }
    }
    return result;
}

void fillMatrix(string key) {
    string all = key;
    for (char c = 'A'; c <= 'Z'; c++) {
        if (c == 'J') continue; // skip J
        if (all.find(c) == string::npos)
            all += c;
    }
    int idx = 0;
    for (int i = 0; i < 5; i++)
        for (int j = 0; j < 5; j++)
            matrix[i][j] = all[idx++];
}

void findPos(char ch, int &row, int &col) {
    if (ch == 'J') ch = 'I';
    for (int i = 0; i < 5; i++)
        for (int j = 0; j < 5; j++)
```

```

        if (matrix[i][j] == ch) {
            row = i;
            col = j;
            return;
        }
    }
}

string encrypt(string text) {
    string pt = "";
    for (char c : text) {
        if (isalpha(c)) {
            c = toupper(c);
            if (c == 'J') c = 'I';
            pt += c;
        }
    }
    if (pt.size() % 2 != 0) pt += 'X'; // make even length

    string ct = "";
    for (int i = 0; i < pt.size(); i += 2) {
        int r1, c1, r2, c2;
        findPos(pt[i], r1, c1);
        findPos(pt[i+1], r2, c2);

        if (r1 == r2) { // same row
            ct += matrix[r1][(c1 + 1) % 5];
            ct += matrix[r2][(c2 + 1) % 5];
        }
        else if (c1 == c2) { // same column
            ct += matrix[(r1 + 1) % 5][c1];
            ct += matrix[(r2 + 1) % 5][c2];
        }
        else { // rectangle
            ct += matrix[r1][c2];
            ct += matrix[r2][c1];
        }
    }
    return ct;
}

int main() {
    string key, plaintext;
    cout << "Enter keyword: ";
    getline(cin, key);

    key = prepareKey(key);

    fillMatrix(key);

    cout << "\nPlayfair Matrix:\n";
    for (int i = 0; i < 5; i++) {
        for (int j = 0; j < 5; j++)

```

```

        cout << matrix[i][j] << " ";
        cout << endl;
    }

    cout << "\nEnter plaintext: ";
    getline(cin, plaintext);

    string ciphertext = encrypt(plaintext);
    cout << "Ciphertext: " << ciphertext << endl;

    return 0;
}

```

❖ Output:

Enter keyword: play

Playfair Matrix:

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

Enter plaintext: games

Ciphertext: EBKFXY

p