**Practical: 04**

**AIM:** Apply the cryptanalytics on Playfair cipher and show how successful it is. Provide the extended version of the Play-Fair cipher. [Hint. Use extended matrix size which incorporates a variety of symbols like alphabets, numbers, special symbols].

**CODE:**

import java.util.\*;

class Prac\_3 {

static char[][] mat5 = new char[5][5], mat10 = new char[10][10];

static String key = "KEYWORD";

static String extSet = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789@#$%^&\*()-+=!?";

// 5x5 Matrix Setup

static void setMat5() {

Set<Character> s = new HashSet<>();

String k = key.toUpperCase().replace("J", "I") + "ABCDEFGHIKLMNOPQRSTUVWXYZ";

int x = 0, y = 0;

for (char c : k.toCharArray()) {

if (!s.contains(c)) {

mat5[x][y] = c;

s.add(c);

if (++y == 5) {

y = 0;

x++;

}

}

}

}

// Encrypt in 5x5 Playfair

static String enc5(String msg) {

msg = msg.toUpperCase().replace("J", "I").replaceAll("[^A-Z]", "");

StringBuilder sb = new StringBuilder();

for (int i = 0; i < msg.length(); i += 2) {

char a = msg.charAt(i), b = (i + 1 < msg.length()) ? msg.charAt(i + 1) : 'X';

if (a == b)

b = 'X';

int[] p1 = getPos5(a), p2 = getPos5(b);

if (p1[0] == p2[0]) {

sb.append(mat5[p1[0]][(p1[1] + 1) % 5]).append(mat5[p2[0]][(p2[1] + 1) % 5]);

} else if (p1[1] == p2[1]) {

sb.append(mat5[(p1[0] + 1) % 5][p1[1]]).append(mat5[(p2[0] + 1) % 5][p2[1]]);

} else {

sb.append(mat5[p1[0]][p2[1]]).append(mat5[p2[0]][p1[1]]);

}

}

return sb.toString();

}

// Get position in 5x5 matrix

static int[] getPos5(char c) {

for (int i = 0; i < 5; i++)

for (int j = 0; j < 5; j++)

if (mat5[i][j] == c)

return new int[] { i, j };

return null;

}

// Attempt pattern-based decryption

static void crack5(String text) {

System.out.println("Cracking 5x5 Cipher... (Needs Frequency Analysis) " + text);

}

// 10x10 Extended Matrix Setup

static void setMat10() {

Set<Character> s = new HashSet<>();

String k = key + extSet;

int x = 0, y = 0;

for (char c : k.toCharArray()) {

if (!s.contains(c)) {

mat10[x][y] = c;

s.add(c);

if (++y == 10) {

y = 0;

x++;

}

}

}

}

// Encrypt in 10x10 Extended Playfair

static String enc10(String msg) {

StringBuilder sb = new StringBuilder();

for (int i = 0; i < msg.length(); i += 2) {

char a = msg.charAt(i), b = (i + 1 < msg.length()) ? msg.charAt(i + 1) : '@';

int[] p1 = getPos10(a), p2 = getPos10(b);

if (p1[0] == p2[0]) {

sb.append(mat10[p1[0]][(p1[1] + 1) % 10]).append(mat10[p2[0]][(p2[1] + 1) % 10]);

} else if (p1[1] == p2[1]) {

sb.append(mat10[(p1[0] + 1) % 10][p1[1]]).append(mat10[(p2[0] + 1) % 10][p2[1]]);

} else {

sb.append(mat10[p1[0]][p2[1]]).append(mat10[p2[0]][p1[1]]);

}

}

return sb.toString();

}

// Get position in 10x10 matrix

static int[] getPos10(char c) {

for (int i = 0; i < 10; i++)

for (int j = 0; j < 10; j++)

if (mat10[i][j] == c)

return new int[] { i, j };

return null;

}

public static void main(String[] args) {

setMat5();

String msg = "HELLO";

String c5 = enc5(msg);

System.out.println("5x5 Playfair Ciphertext: " + c5);

crack5(c5);

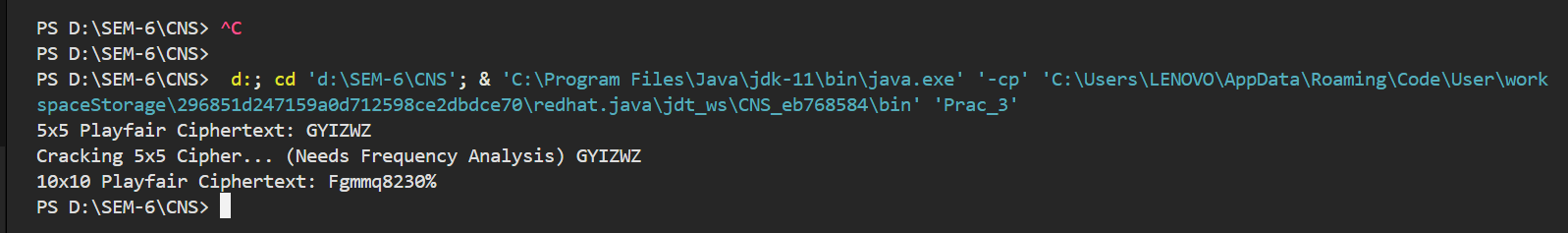
setMat10();

String c10 = enc10("Hello@123");

System.out.println("10x10 Playfair Ciphertext: " + c10);

}

}

**OUTPUT:**