Information Security & Cryptography

Nehal Jhajharia
Lab Assignment 4

1)

```
// Implement columnar transposition cipher.

import java.util.*;

public class Columnar {
    static Scanner input = new Scanner(System.in);
    static String key = new String();
    static String plaintext = new String();
    static String ciphertext = new String();
    static char sortedKey[];
    static int sortedKeyPos[];

    static void setInputs() {
        System.out.print("Enter key : ");
        key = input.nextLine();
        System.out.print("Enter text : ");
        plaintext = input.nextLine();

        sortedKeyPos = new int[key.length()];
        sortedKey = key.toCharArray();
        process();
    }

    static void process() {
        int min = 0;
        int i = 0;
        int j = 0;
        char orginalKey[] = key.toCharArray();
        char temp = '\0';

        for (i = 0; i < key.length(); i++) {
            min = i;
            for (j = i; j < key.length(); j++) {
                if (sortedKey[min] > sortedKey[j]) {
                    min = j;
                }
            }

            if (min != i) {
                temp = sortedKey[i];
                sortedKey[i] = sortedKey[min];
                sortedKey[min] = temp;
            }
        }

        System.out.println(sortedKey);

        for (i = 0; i < key.length(); i++) {
            for (j = 0; j < key.length(); j++) {
                if (orginalKey[i] == sortedKey[j]) {
                    sortedKeyPos[i] = j;
                }
            }
        }

        System.out.println(Arrays.toString(sortedKeyPos));
    }

    static String encrypt() {
        int i = 0;
        int j = 0;

        int row = plaintext.length() / key.length();
        int extrabit = plaintext.length() % key.length();
        int exrow = (extrabit == 0) ? 0 : 1;
        int coltemp = -1;
        int totallen = (row + exrow) * key.length();
        char pmat[][] = new char[(row + exrow)][(key.length())];
        char encry[] = new char[totallen];

        row = 0;

        for (i = 0; i < totallen; i++) {
```

```java
        coltemp++;
        if (i < plaintext.length()) {
            if (coltemp == (key.length())) {
                row++;
                coltemp = 0;
            }
            pmat[row][coltemp] = plaintext.charAt(i);
        }

        else {
            pmat[row][coltemp] = '-';
        }
    }

    int len = -1, k;

    for (i = 0; i < key.length(); i++) {
        for (k = 0; k < key.length(); k++) {
            if (i == sortedKeyPos[k]) {
                break;
            }
        }
        for (j = 0; j <= row; j++) {
            len++;
            encry[len] = pmat[j][k];
        }
    }

    String p1 = new String(encry);
    ciphertext = p1;
    return (new String(p1));
}

static String decrypt() {
    int i = 0;
    int j = 0;
    int k = 0;
    char encry[] = ciphertext.toCharArray();

    int col = key.length();
    int row = ciphertext.length() / col;
    char pmat[][] = new char[row][col];
    int tempcnt = -1;

    for (i = 0; i < col; i++) {
        for (k = 0; k < col; k++) {
            if (i == sortedKeyPos[k]) {
                break;
            }
        }

        for (j = 0; j < row; j++) {
            tempcnt++;
            pmat[j][k] = encry[tempcnt];
        }
    }

    printMatrix(pmat);

    char p1[] = new char[row * col];
    k = 0;
    for (i = 0; i < row; i++) {
        for (j = 0; j < col; j++) {
            if (pmat[i][j] != '-') {
                p1[k++] = pmat[i][j];
            }
        }
    }

    return (new String(p1));
}

static void printMatrix(char matrix[][]) {
    for (int i = 0; i < matrix.length; i++) {
        System.out.print(" | ");
        for (int j = 0; j < matrix[0].length; j++) {
            System.out.print(matrix[i][j]);
            System.out.print(" | ");
        }
        System.out.println();
    }
}
```

```
    public static void main(String[] args) {
        setInputs();

        System.out.println("Cipher Text : " + encrypt());
        System.out.println("Decrypted Text : " + decrypt());
    }
}
```

```
jhajharia@Nehals-MacBook-Air Asmt4 % Java Columnar
Enter key : nehal
Enter text : nehal jhajharia
aehln
[4, 1, 2, 0, 3]
Cipher Text : aaiejahhrljan h
 | n | e | h | a | l |
 |   | j | h | a | j |
 | h | a | r | i | a |
Decrypted Text : nehal jhajharia
jhajharia@Nehals-MacBook-Air Asmt4 %
```

2)

```
// Implement Vernam cipher.

import java.util.*;

public class Vernam {
    static Scanner input = new Scanner(System.in);
    static String key = new String();
    static String plaintext = new String();
    static String ciphertext = new String();

    public static String encrypt(String text, String key) {
        String cipherText = "";
        int cipher[] = new int[key.length()];

        for (int i = 0; i < key.length(); i++) {
            cipher[i] = text.charAt(i) - 'A' + key.charAt(i) - 'A';
        }

        for (int i = 0; i < key.length(); i++) {
            if (cipher[i] > 25) {
                cipher[i] = cipher[i] - 26;
            }
        }

        for (int i = 0; i < key.length(); i++) {
            int x = cipher[i] + 'A';
            cipherText += (char) x;
        }

        return cipherText;
    }

    public static String decrypt(String s) {
        String text = "";

        int plain[] = new int[key.length()];

        for (int i = 0; i < key.length(); i++) {
            plain[i] = s.charAt(i) - 'A' - (key.charAt(i) - 'A');
        }

        for (int i = 0; i < key.length(); i++) {
            if (plain[i] < 0) {
                plain[i] = plain[i] + 26;
            }
        }

        for (int i = 0; i < key.length(); i++) {
            int x = plain[i] + 'A';
            text += (char) x;
        }

        return text;
    }

    public static void main(String[] args) {
```

```java
        System.out.print("Enter key : ");
        key = input.nextLine().toUpperCase();
        System.out.print("Enter text : ");
        plaintext = input.nextLine().toUpperCase();

        ciphertext = encrypt(plaintext, key);

        System.out.println("Cipher Text : " + ciphertext);
        System.out.println("Decrypted Text : " + decrypt(ciphertext));
    }
}
```

```
jhajharia@Nehals-MacBook-Air Asmt4 % javac Vernam.java
jhajharia@Nehals-MacBook-Air Asmt4 % Java Vernam
Enter key : nehal
Enter text : apple
Cipher Text : NTWLP
Decrypted Text : APPLE
jhajharia@Nehals-MacBook-Air Asmt4 %
```