12. random keys

import java.util.Random;

public class VigenereCipher {

public static void main(String[] args) {

String plaintext = "HELLO WORLD";

int[] key = generateKey(plaintext.length());

System.out.println("Plaintext: " + plaintext);

String ciphertext = encrypt(plaintext, key);

System.out.println("Ciphertext: " + ciphertext);

String decryptedText = decrypt(ciphertext, key);

System.out.println("Decrypted text: " + decryptedText);

}

public static int[] generateKey(int length) {

int[] key = new int[length];

Random random = new Random();

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

key[i] = random.nextInt(26) + 1;

}

return key;

}

public static String encrypt(String plaintext, int[] key) {

String ciphertext = "";

int keyIndex = 0;

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

char c = plaintext.charAt(i);

int shift = key[keyIndex];

char encryptedChar = shiftChar(c, shift);

ciphertext += encryptedChar;

keyIndex = (keyIndex + 1) % key.length;

}

return ciphertext;

}

public static String decrypt(String ciphertext, int[] key) {

String decryptedText = "";

int keyIndex = 0;

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

char c = ciphertext.charAt(i);

int shift = key[keyIndex];

char decryptedChar = shiftChar(c, -shift);

decryptedText += decryptedChar;

keyIndex = (keyIndex + 1) % key.length;

}

return decryptedText;

}

public static char shiftChar(char c, int shift) {

if (!Character.isLetter(c)) {

return c;

}

int base = Character.isLowerCase(c) ? 'a' : 'A';

int offset = c - base;

int shiftedOffset = (offset + shift + 26) % 26;

return (char) (base + shiftedOffset);

}

}