Ministerul Educatiei, Culturii și Cercetarii al Republicii Moldova

Universitatea Tehnică a Moldovei

Facultatea Calculatoare, Informatică şi Microelectronică

Departamentul Ingineria Software și Automatica

Raport

For laboratory Nr. 1

At the course „ Cryptographic methods of information protection"

Student:

Prodan Denis FAF-211

Verifier:

Aureliu ZGUREANU

Chișinău – 2023

Subject: Caesar's Cipher

Objectives:

1. To implement the Caesar algorithm for the English alphabet in one of the languages programming. Use only the letter encoding as shown in table 1 (not set allows to use encodings specified in the programming language, e.g. its ASCII Unicode). Key values will be between 1 and 25 inclusive and no other values are allowed

The values of the text characters are between 'A' and 'Z', 'a' and 'z' and there are no other premises values. If the user enters other values - the correct tuning will be suggested.

Before encryption, the text will be converted to uppercase and spaces will be removed

The user will be able to choose the operation - encryption or decryption, he will be able to enter the key the message or the cryptogram will get respectively the cryptogram or the decrypted message

2. To implement the Caesar algorithm with 2 keys, keeping the conditions expressed above Task 1. In addition, key 2 must contain only letters of the Latin alphabet, and have length not less than 7.

Caesar's Cipher

In this cipher each letter of the plaintext is replaced by a new letter obtained by move alphabetically. The secret key k, which is the same for both encryption and decryption consists of the number indicating the alphabetical displacement, i.e. $k \in \{1, 2, 3, \ldots, n-1\}$, where n is the length of the alphabet. The encryption and decryption of the message with the Caesar cipher can be defined though FORMULATIONS

$$c = ek(x) = x + k \pmod{n},$$
$$m = dk(y) = y - k \pmod{n},$$

where x and y are the numerical representation of the respective character of the clear text. I'm working
called Modulo (a mod b) returns the remainder of dividing the integer a by the integer t
b. For example, for k = 3 we have (fig. 1):

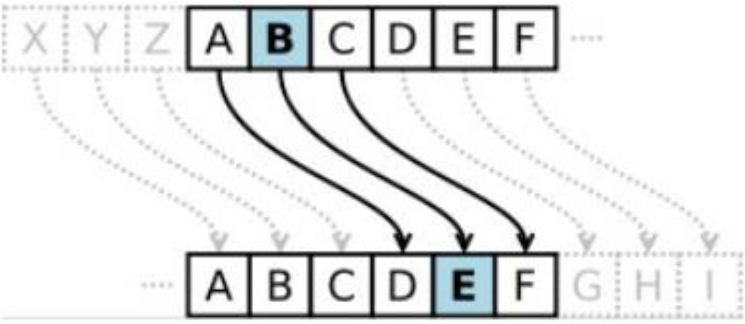


Figure 1 – Example of alphabetical movement

To increase the cryptoresistance of the Caesar cipher, a permutation of the alphabet can be applied by applying a keyword (not to be confused with the base key of the cipher). This has key can be any sequence of letters of the alphabet - either a vocabulary word or one without meaning.

Let the second key be k2 =cryptography. I apply this key to the alphabet

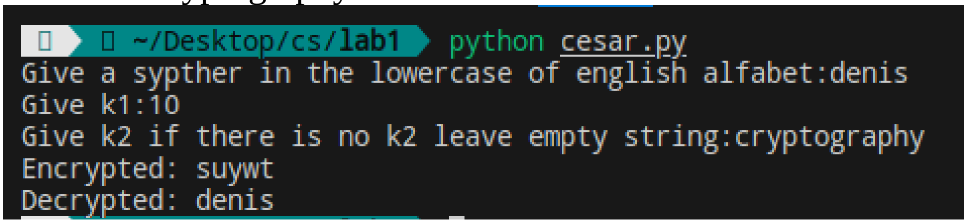
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

and we get:

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

This new order I obtained by placing the letters of kă ă ţ 2 at the beginning, then follows the other letters of the alphabet in their natural order. We will take into account the fact that the letters are new will be repeated, i.e. if the letter occurs several times, it is placed only once given.

Results:
In this example I used lower case english alfabet, the first key is 10 and the second key for permutation is the word cryptography.



```
 ~/Desktop/cs/lab1  python cesar.py
Give a sypther in the lowercase of english alfabet:denis
Give k1:10
Give k2 if there is no k2 leave empty string:cryptography
Encrypted: suywt
Decrypted: denis
```

FIGURE 2

Conclusion: In this laboratory work I learned the principle of operation like this Caesar's encryption algorithm. I made an application that uses the cipher of Cezar cut one and two keys to encrypt and decrypt messages. Even if the version with a douche chei doesn't seem so complicated at first glance, but it significantly decreases the chances of such and such breaks this cipher, compared to the one with only one key. If we used the exhaustive method we will need to check 26! * 25 options to decode the message.