**Ministerul Educaţiei și Cercetării al Republicii Moldova**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică și Microelectronică**

**Departamentul Ingineria Software și Automatica**

Report

Laboratory work Nr.1

“Cryptography and Security”

Elaborated: st. gr. FAF-213 Cojuhari Milena

Verified: asist. univ. Mîțu Cătălin

Chişinău – 2023

**Subject:** Caesar Cipher

**Tasks:**

1 To implement the Caesar cipher algorithm for the English language alphabet in one of the programming languages. Use only the letter encoding as shown in Table 1 (using the specific encodings specified in the programming language, such as ASCII or Unicode, is not allowed). Key values will be in the range of 1 to 25, inclusive, and no other values are allowed. The values of text characters should be between 'A' and 'Z', 'a' and 'z', and no other values are allowed. If the user enters other values, they will be prompted with the correct range. 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, enter the key, message, or ciphertext, and obtain the corresponding ciphertext or decrypted message.

2 To implement the Caesar cipher algorithm with 2 keys while maintaining the conditions expressed in Task 1. Additionally, key 2 must consist of only Latin alphabet letters and have a length of no less than 7.

**Caesar Cipher**

The Caesar Cipher. In this cipher, each letter of the plaintext is replaced with a new letter obtained by an alphabetic shift. The secret key "k," which is the same for both encryption and decryption, represents the number that indicates the alphabetic shift, i.e., k ∈ {1, 2, 3, ..., n–1}, where "n" is the length of the alphabet. The encryption and decryption of the message with the Caesar cipher can be defined by the formulas:

c = ek(x) = x + k (mod n)

m = dk(y) = y – k (mod n)

Here, "x" and "y" represent the numerical representation of the corresponding character in the plaintext "m" and the ciphertext "c." The function called Modulo (a mod b) returns the remainder of dividing the integer "a" by the integer "b." This encryption method is named after Julius Caesar, who used it to communicate with his generals, using the key k = 3 (Table 1).

For example, with k = 3:

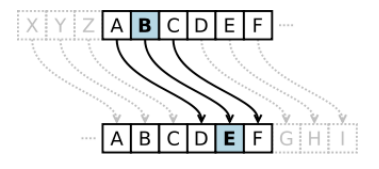
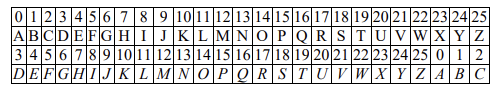


Fig 1. Example of an alphabetic shift.

ek(S) = 18 + 3 (mod 26) = 21 = V

dk(V) = 21 – 3 (mod 26) = 18 = S

In this case, for the plaintext "cifrul cezar," we obtain the ciphertext "fliuxo fhcdu".



Tab 1. Caesar Cipher with key k=3

The Caesar Cipher is very easy to break, making it a very weak cipher. As a result, a cryptanalyst can obtain the plaintext by trying all 25 possible keys. It is not known how useful the Caesar cipher was during the time when it was used by the individual from whom it derives its name, but it is likely that it was reasonably secure, as only a few of Caesar's enemies were capable of writing and reading, let alone having knowledge of cryptanalysis concepts.

**RESULTS**

At the run of the program, a message that asks the user to choose between encrypting or decrypting a text will be displayed.



Fig 2. Choosing Encryption of Decryption

Next, the user will be asked to choose between the execution of the classical Caesar Cipher method (with 1 key) or the method with 2 keys.



Fig 3. Choosing the 1-key or 2-key method

In dependence of the method selected, the user will be asked to enter the text, the first key and, if necessary, the second key.

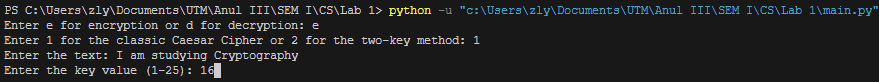


Fig 4. Introducing the data

In the end the result, in this case the encrypt text will be displayed.

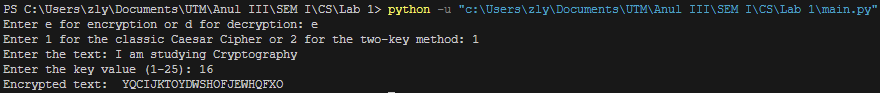


Fig 5. Encryption with 1 key

Example of the decryption of the text priorly obtained.

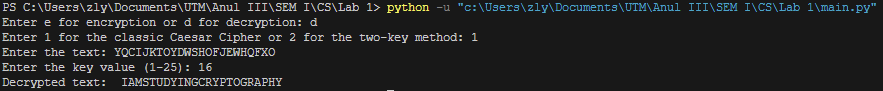


Fig 6. Decryption with 1 key

Here there is also an example of the encryption method with two keys. To see better the differences between the 2 methods I used the same plaintext and first-key.

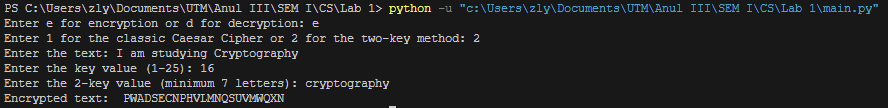


Fig 7. Encryption with 2 keys

And the decrypted text obtained from the answer of the last result.

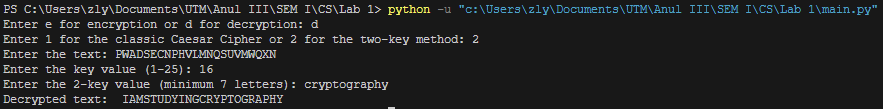


Fig 8. Decryption with 2 keys

Some examples of what happens when the introduced data are wrong (the user will be asked to enter the data again):

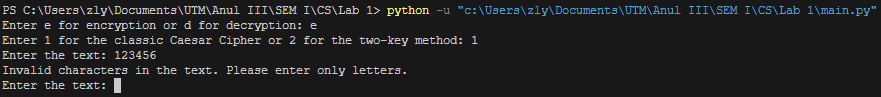


Fig 9. Introducing invalid characters in the plaintext

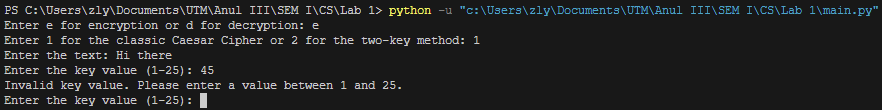


Fig 10. Giving key 1 a value outside the range

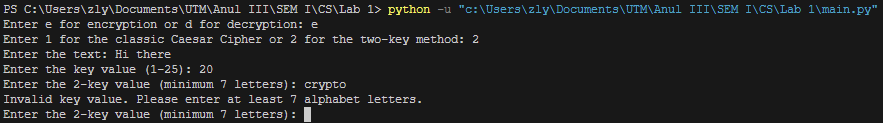


Fig 11. Introducing less than 7 letters for key 2

**CONCLUSION**

In this laboratory work, I implemented the Caesar Cipher algorithm with one or two key features. The 1-key method although easy to understand and code does not give much security to the encrypted message. However, the 2-key method enhance the encryption complexity and offers a much more secure encryption.

This project provided me with first experience in cryptography and with a layer foundation for further exploration in this field.

**GitHub LINK**