Ministerul Educaţiei și Cercetării al Republicii Moldova

Universitatea Tehnică a Moldovei

Facultatea Calculatoare, Informatică și Microelectronică

Report

Laboratory Work nr. 1

Caesar Cypher

Elaborated by: Afteni Maria, FAF-213

Verified by: Cătălin Mîțu

Chișinău – 2023

Subject: Caeser Cypher

Tasks:

1. To implement the Caesar algorithm for the English alphabet in one of the programming languages. Use only the letter encoding as shown in table 1 (it is not allowed to use the encodings specified in the programming language, eg ASCII or Unicode). Key values will be between 1 and 25 inclusive and no other values are allowed. The text character values are between 'A' and 'Z', 'a' and 'z' and no other values are assumed. 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, will be able to enter the key, message or cryptogram and get the decrypted cryptogram or message respectively.

2. To implement the Caesar algorithm with 2 keys, keeping the conditions expressed in Task 1. In addition, key 2 must contain only letters of the Latin alphabet, length no less than 7.

Caesar Cypher:

The Caesar cipher, alternatively referred to as the Caesar shift or Caesar code, is a specific type of substitution cipher. It takes its name from Julius Caesar, who is historically associated with its use for safeguarding confidential messages. In this encryption technique, each letter in the original text undergoes a fixed displacement up or down the alphabet, determined by a numerical value known as the "key."

In the process of utilizing the Caesar cipher, the first step is to select a key, which is a positive integer. This key serves as the guiding factor for shifting each letter within the plaintext across the alphabet. For instance, if the key is set to 3, 'A' is transformed into 'D,' 'B' into 'E,' and so forth.

Encryption takes place by advancing each letter in the plaintext forward through the alphabet by the specified key value. In cases where you reach the end of the alphabet, a wraparound occurs. As an example, employing a key of 3 results in "HELLO" becoming "KHOOR."

Decryption, conversely, involves reversing the process by shifting each letter in the ciphertext backward through the alphabet by the identical key value. In this way, using the key of 3, "KHOOR" is decrypted back to its original form, "HELLO."

Results:

When the program is executes, in the terminal is outputted an interactive menu, where the user can choose what service he wants to execute. The menu will appear after each service, once it is completed.

```
1. Encrypt with numerical key
2. Decrypt with numerical key
3. Encrypt with key word
4. Decrypt with key word
5. Exit
Chose service:
```

Fig 1. Options Menu

The user can choose one of the 4 services provided. The first two are the "Encrypt with numerical key" and "Decrypt with numerical key". When this options are selected, the user is asked to input the message that will be encrypted or decrypted with the Caeser Cypher and the numerical key.

```
Chose service: 1
Input the message: Hello
Input the key: 4
Encrypted message LIPPS
```

Fig 2. "Encrypt with numerical key" service

In both, encryption and decryption, the input strings are verified, if the strings contain only letters. But firstly, the strings are going through a function that eliminates all the white spaces from them.

```
Chose service: 2
Input the encoded message: LIPPS
Input the key: 4
Decrypted message: HELLO
```

Fig 3. "Decrypt with numerical key" service

The next options are the encryption and decryption with a second key, that is a word. The key word changes the arrangements of the letters and creates a more secure encryption.

```
Chose service: 3
Input the message: Hello
Input the key word: cryptography
Input the numerical key: 4
Encrypted message with key word: FKSSB
```

Fig 4. "Encrypt with word key" service

The decryption action is done similar to the encryption one.

```
Chose service: 4
Input the encoded message: FKSSB
Input the key word: cryptography
Input the numerical key: 4
Decrypted message with key word: HELLO
```

Fig 5. “Decrypt with word key” service

The key word should have a length equal or bigger that 7. If the condition is not respected, the user is asked to input another key word.

```
Chose service: 4
Input the encoded message: Hello
Input the key word: crypto
Invalid key word
Input the key word: cryptogra
Input the numerical key: 4
Decrypted message with key word: BGHHR
```

Fig 6. Invalid key word case

To exit the program, the user should select the option 5.

```
Chose service: 5
Exiting the program
```

Fig 7. Exiting the program

Conclusions:

In conclusion, the laboratory work on implementing the Caesar cipher with letter encoding and Caesar cipher permutations has been an insightful learning experience. Through this project, I gained a deeper understanding of fundamental cryptographic principles and encryption techniques.

The Caesar cipher, despite its simplicity, provided valuable insights into the concept of substitution ciphers and how they can be applied to secure information. It was intriguing to see how a basic shift in the alphabet could transform plaintext into ciphertext, and vice versa.