

Ministry of Education, Culture and Research of the Republic of Moldova

Technical University of Moldova

Department of Software and Automation Engineering

**REPORT**

Laboratory work No. 3

**Discipline**: Cryptography and Security

Elaborated: Cravcenco Dmitrii FAF-223,

Checked: Dumitru Nirca asist. univ.,

Chișinău 2024

## 

## Topic: Vigenère Cipher

## Tasks:

Реализовать алгоритм Виженера на одном из языков программирования для сообщений на русском языке (33 буквы). Буквы кодируются числами 0, 1, ... 32. Значения символов сообщения находятся между «A» и 'Z', 'a' и 'z' и никакие другие значения не допускаются. Если пользователь введет другие значения - будет предложен правильный диапазон символов. Длина ключа не должна быть меньше 7. Шифрование и дешифрование будет производиться по формулам математической модели, представленной выше. В сообщении сначала нужно убрать пробелы, потом все буквы будут заменены на заглавные. Пользователь сможет выбрать операцию - шифрование или дешифрование, сможет ввести ключ, сообщение или криптограмму и получит криптограмму или расшифрованное сообщение.

## Theoretical notes:

The Vigenère cipher is a classic method of encrypting alphabetic text by using a simple form of polyalphabetic substitution. It's named after the French cryptologist Blaise de Vigenère, though it was actually described earlier by Giovan Battista Bellaso.

How It Works:

Key: The encryption key is a string of letters. This key is repeated as needed to match the length of the plaintext message.

Encryption: Each letter in the plaintext is shifted along some number of places, determined by the corresponding letter in the repeated key.

Formula: E(x) = (P(x) + K(x)) % 26

Where E(x) is the encrypted letter, P(x) is the plaintext letter, and K(x) is the key letter.

Decryption: The reverse process is applied, using the same key.

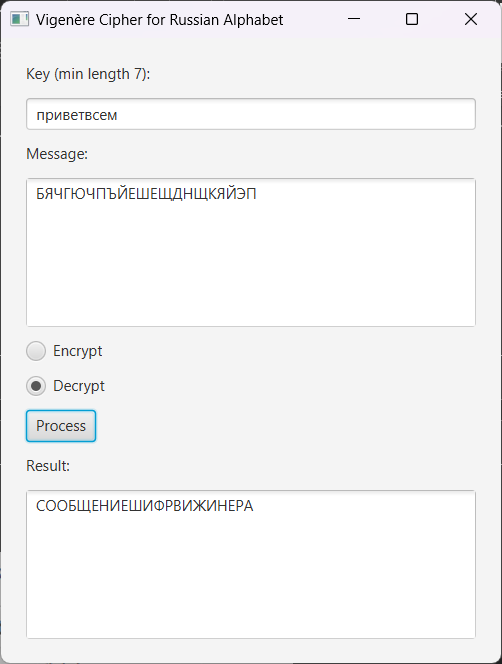
Formula: D(x) = (E(x) - K(x) + 26) % 26

Where D(x) is the decrypted letter.

**Implementation(Var. Nr.2)**

Code is on GitHub: https://github.com/DmitriiKaban/Cryptography-Labs

So here is UI:



**Fig.1**: UI

Here is logic behind encryption:

*private String encryptMessage(String key, String message) {  
 StringBuilder encryptedText = new StringBuilder();  
 String preparedKey = prepareKey(key, message.length());  
  
 for (int i = 0; i < message.length(); i++) {  
 char messageChar = message.charAt(i);  
 char keyChar = preparedKey.charAt(i);  
  
 int messageIndex = RUSSIAN\_ALPHABET.indexOf(messageChar);  
 int keyIndex = RUSSIAN\_ALPHABET.indexOf(keyChar);  
  
 int cipherIndex = (messageIndex + keyIndex) % ALPHABET\_SIZE;  
 encryptedText.append(RUSSIAN\_ALPHABET.charAt(cipherIndex));  
 }  
  
 return encryptedText.toString();  
}*

So, it basically implements modular arithmetic operations, it sums up indexes of the current message’s character and key’s index character and computes the remainer when dividing by the number of letters in the alphabet.

Here is logic behind decryption process:  
private String decryptMessage(String key, String message) {  
 StringBuilder decryptedText = new StringBuilder();  
 String preparedKey = prepareKey(key, message.length());  
  
 for (int i = 0; i < message.length(); i++) {  
 char messageChar = message.charAt(i);  
 char keyChar = preparedKey.charAt(i);  
  
 int messageIndex = *RUSSIAN\_ALPHABET*.indexOf(messageChar);  
 int keyIndex = *RUSSIAN\_ALPHABET*.indexOf(keyChar);  
  
 int decipherIndex = (messageIndex - keyIndex + *ALPHABET\_SIZE*) % *ALPHABET\_SIZE*;  
 decryptedText.append(*RUSSIAN\_ALPHABET*.charAt(decipherIndex));  
 }  
  
 return decryptedText.toString();  
}

Basically it is the same, instead of summing up we subtract.

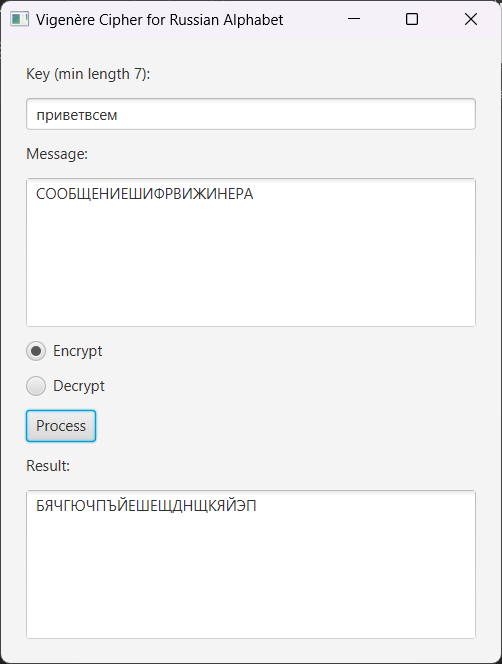


Figure 2 The process of ciphering of a message

Conclusion:

In this laboratory work, I successfully developed a JavaFX application to implement Vigenère cipher. The key components of the application include:

1. **User Interface Setup**:
   1. Created text areas for inputting encrypted text and displaying decrypted text.
   2. Designed radio buttons for UX
2. **Vigenère cipher:**
   1. Implemented cipher encryption using modular arithmetic
   2. Implemented cipher decryption using modular arithmetic