

MINISTRY OF EDUCATION, CULTURE AND RESEARCH OF REPUBLIC OF MOLDOVA
TECHNICAL UNIVERSITY OF MOLDOVA
FACULTY OF COMPUTERS, INFORMATICS AND MICROELECTRONICS
DEPARTMENT OF SOFTWARE ENGINEERING AND AUTOMATICS

Cryptography and Security

Laboratory work 3: Polialphabetical Chiper

Elaborated:

st.gr. FAF-211

Gazea Sandu

Verified:

asist.univ.

Cătălin Mîțu

Chișinău, 2023

Vigenère cipher

Historical Context

The Vigenère cipher, named after Blaise de Vigenère, a French diplomat and cryptographer who lived in the 16th century, is a method of encrypting alphabetic text through a simple form of polyalphabetic substitution. Although Vigenère did not invent this cipher, it was wrongly attributed to him, and the misnomer has persisted throughout history.

This cipher gained notoriety for the level of security it provided, earning it the nickname “le chiffre indéchiffrable,” or “the unbreakable cipher.” During the Renaissance period, the Vigenère cipher was considered highly secure and was widely used to protect sensitive information from unauthorized access. However, with the advent of computational power and modern cryptographic techniques, the Vigenère cipher is now easy to crack and mainly serves an educational purpose to introduce concepts of cryptography.

Mechanism of the Vigenère Cipher

The Vigenère cipher employs a form of polyalphabetic substitution, where each letter in the plaintext is shifted along the alphabet by a number of positions defined by a keyword. The keyword is repeated or truncated to match the length of the plaintext. Each letter of the keyword denotes a shift value corresponding to its position in the alphabet, making the cipher polyalphabetic.

Encryption Process

1. **Plaintext:** The original, unencrypted message.
2. **Key:** A word or phrase repeated to the length of the plaintext.
3. **Encryption:** Each letter in the plaintext is shifted according to the corresponding letter in the key.

The mathematical representation for the encryption of a plaintext letter P with a key letter K into a ciphertext letter C can be expressed as

$$C \equiv (P + K) \pmod{26}$$

Decryption Process

Decryption employs the reverse operation, subtracting the key letter's value from the ciphertext letter's value. Mathematically, the decryption can be expressed as

$$P \equiv (C - K) \pmod{26}$$

Implementation

Task 3.2

Implement the Vigenere algorithm in one of the programming languages for messages in Romanian (31 letters), these being encoded with the numbers 0, 1, ... 30. The values of the text characters are between 'A' and 'Z', 'a' and 'z', and no other values are allowed. If the user enters other values, they will be suggested the correct range of characters. The length of the key should not be less than 7. Encryption and decryption will be carried out in accordance with the formulas from the mathematical model presented above. In the message, spaces must first be eliminated, then all letters should be converted to uppercase. The user can choose the operation - encryption or decryption, can enter the key, the message or the cryptogram, and will obtain the cryptogram or the decrypted message.

Section 1: Definitions and Initializations

The code initiates by delineating the Romanian alphabet, a string containing 31 distinctive characters. This predefined set of characters is employed in subsequent stages for both the validation of input and the processes of encryption and decryption.

Section 2: Input Validation

The function `validate_input` is tasked with the verification of each character within the supplied text to ensure their existence within the predefined Romanian alphabet. The initiation of an error message occurs if an outlier character, one not belonging to the alphabet, is detected. This measure ensures the processing of only appropriate and valid characters.

Section 3: Character Encoding and Decoding

In this section, the `encode_char` function is introduced, converting a specific character into its respective numerical index within the Romanian alphabet. Conversely, the `decode_char` function reverts this process, translating a numerical index back into its associated character.

Section 4: Vigenère Encryption

The encryption process is executed within the `vigenere_encrypt` function. Initial steps involve the validation of both the plaintext and key, ensuring the presence of valid characters exclusively. Following the elimination of spaces and the conversion of all characters to uppercase form, an iterative process begins. Each character of the plaintext undergoes a shift, influenced by the key, leading to the encryption of the message. Additionally, the shifted alphabet, influenced by the key, is calculated and displayed.

Section 5: Alphabet Shifting Based on Key

The function `shift_alphabet_based_on_key` is responsible for computing a modified version of the alphabet, influenced by the encryption key. This section elucidates the reorganization of the alphabet occurring during the encryption phase.

Section 6: Vigenère Decryption

The decryption process is facilitated through the `vigenere.decrypt` function. Mirroring the encryption function, initial phases involve the validation of both the ciphertext and key. Each character of the ciphertext is then subjected to a reverse shift, influenced by the key, unveiling the original plaintext message.

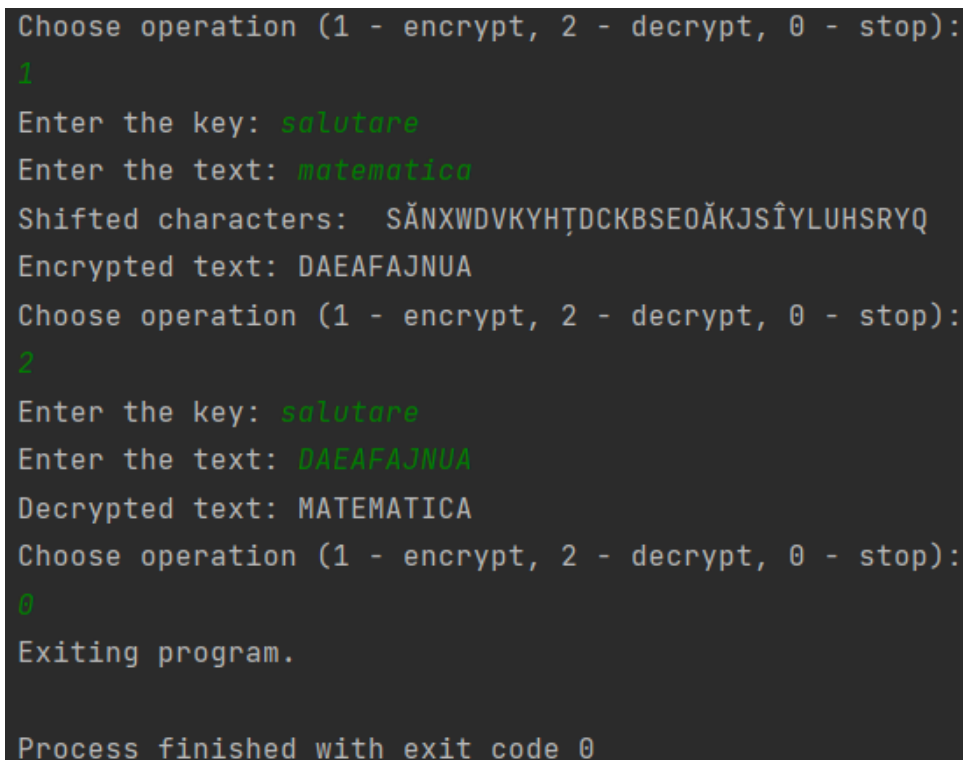
Section 7: User Interaction

A loop is initiated, prompting users to opt between encryption, decryption, or program termination. Contingent on the user's selection, prompts appear for the entry of a key and either a plaintext (for encryption purposes) or a ciphertext (for decryption). Subsequently, the appropriate function is invoked, displaying the resultant output.

The complete code is accessible on my GitHub, attached below for further review and analysis.

[<https://github.com/GSandu1/CS.git>]

Below is a screenshot displaying the output generated by the implemented Vigenère cipher code.



```
Choose operation (1 - encrypt, 2 - decrypt, 0 - stop):
1
Enter the key: salutare
Enter the text: matematica
Shifted characters: SĂNXWDVKYHȚDCKBSEOĂKJSÎYLUHSRYQ
Encrypted text: DAEAFJNUA
Choose operation (1 - encrypt, 2 - decrypt, 0 - stop):
2
Enter the key: salutare
Enter the text: DAEAFJNUA
Decrypted text: MATEMATICA
Choose operation (1 - encrypt, 2 - decrypt, 0 - stop):
0
Exiting program.

Process finished with exit code 0
```

Task 3.2

Conclusion

In the multifaceted domain of cryptography, past encryptions like the Vigenere cipher and Playfair algorithm are monumental, serving as cornerstones for today's advanced cryptographic systems. The Vigenere cipher, often misattributed to Blaise de Vigenere, augmented security by utilizing multiple shifts, thus complicating frequency analysis. On the other hand, the Playfair algorithm, the brainchild of Charles Wheatstone but named after Baron Lyon Playfair, revolutionized encryption by encoding pairs of letters instead of single characters, enhancing communication security during wars.

Though these methods have fallen into obsolescence given the advancements in computational power and cryptographic sophistication, their significance cannot be overstated. In today's era of quantum computing, where encryption and security are paramount, the ingenuity of historical ciphers like Vigenere and Playfair offer profound insights. They embody the dynamic evolution of cryptography, echoing the persistent human effort to augment privacy and security amidst ever-evolving technological advancements.

Every cutting-edge cryptographic technique and security protocol today is rooted in the trials, innovations, and accomplishments of earlier encryption methods. While not directly applied in contemporary contexts, Vigenere and Playfair ciphers remain instrumental, offering foundational knowledge that enhances the understanding and development of emerging cryptographic and cybersecurity innovations. They are emblematic of the perpetual progression in the ongoing quest to safeguard sensitive information and secure communications in an increasingly connected world.