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

Criptography and Security

Laboratory work 3: Criptanaliza cifrurilor monoalfabetice

Elaborated:
st.gr. FAF-212 Cristian Brinza
Verified:
asist.univ. Cătălin Mîțu

Chișinău, 2023

Introduction

The Vigenere Cipher is a method of encrypting alphabetic text by using a polyalphabetic substitution mechanism. It uses a keyword to shift letters in the plaintext to produce the ciphertext. This laboratory work focuses on implementing this cipher with specific constraints, such as handling the Romanian alphabet, ensuring a minimum key length, and providing a user-friendly interface for encryption and decryption.

3.1. Cifrul Vigenère Metoda de criptare cunoscută sub numele de „cifrul Vigenère” a fost atribuită greşit lui Blaise de Vigenère în secolul al XIX-lea și, de fapt, a fost descrisă pentru prima dată de Giovan Battista Bellaso în cartea sa din 1553 La cifra del. Sig. Vigenère a creat un cifru asemănător, dar totuși diferit și mai puternic în 1586. Pe da altă parte, cifrul Vigenere folosește aceleași operații ca și cifrul Cezar. Cifrul Vigenere și fel deplasează literele, dar, spre deosebire de Cezar, nu se poate sparge uşor în 26 combinații. Cifrul Vigenere folosește o deplasare multiplă. Cheia nu este constituită de o singură deplasare, ci de mai multe, fiind generate de câțiva întregi ki, unde $0 \leq ki \leq 25$, dacă luăm ca reper alfabetul latin cu 26 de litere.

Sarcină 3.2.

De implementat algoritmul Vigenere în unul din limbajele de programare pentru mesaje în limba română (31 de litere), acestea fiind codificate cu numerele 0, 1, ... 30. Valorile caracterelor textului sunt cuprinse între ‘A’ și ’Z’, ’a’ și ’z’ și nu sunt premise alte valori. În cazul în care utilizatorul introduce alte valori - i se va sugera diapazonul corect al caracterelor. Lungimea cheii nu trebuie să fie mai mică de 7. Criptarea și decriptarea se va realiza în conformitate cu formulele din modelul matematic prezentat mai sus. În mesaj mai întâi trebuie eliminate spațiile, apoi toate literele se vor transforma în majuscule. Utilizatorul va putea alege operația - criptare sau decriptare, va putea introduce cheia, mesajul sau criptograma și va obține criptograma sau mesajul decriptat.

Methodology

1. Alphabet and Matrix Initialization
The Romanian alphabet is initialized and a Vigenere matrix is created by shifting the alphabet characters for each row.

2. Key and Text Processing
The key is extended to match the length of the input text, and both are converted to their corresponding indexes in the alphabet.

3. Input Validation
Validation ensures that only valid characters from the specified alphabet are used and that the key length adheres to the specified minimum length.

4. Encryption and Decryption
The cipher operation is performed using the Vigenere matrix, and the result is produced based on the operation type (encryption/decryption) selected by the user.

5. User Interaction
A menu-driven interface allows users to interact with the program, choose operations, and input data.

Implementation
The implementation involves creating a VigenereCipher class that encapsulates all cipher-related functionality and a menu function that provides a user-friendly interface for interaction.

Key methods and functionalities include:

_create_matrix: Generates the Vigenere matrix.

_extend_key: Extends the key to match the input text length.
_get_indexes: Converts characters to their corresponding alphabet indexes.
encrypt_decrypt: Performs the encryption or decryption based on user input.
Results

The implementation successfully encrypts and decrypts messages using the Vigenere Cipher, adhering to the specified constraints and requirements. It validates inputs, processes the key and text, and provides a user-friendly interface for interaction.

Python

This repository contains a Python implementation of the Vigenere Cipher, specifically tailored for messages in the Romanian language, adhering to specific constraints and requirements.

Overview

The Vigenere Cipher is a method of encrypting alphabetic text by using a simple form of polyalphabetic substitution. This implementation provides a class VigenereCipher that encapsulates all the functionality related to the cipher and a user-friendly menu for interaction.

Features

- **Alphabet Specification**: Utilizes the Romanian alphabet, which contains specific characters.
- **Key Length Validation**: Ensures the key length is at least 7.
- **Character Validation**: Validates that only valid characters from the specified alphabet are used.
- **Uppercase Conversion and Space Removal**: Converts all characters to uppercase and removes spaces.
- **User Interaction**: Provides a menu for user-friendly interaction and operation selection.

Class Methods Explanation

1. Initialization and Matrix Creation

Upon initialization, the class takes an alphabet as input, converts it to uppercase, and creates the Vigenere matrix, essential for the cipher operation.

2. Shifting Rows and Matrix Creation

The `_shift_row` method shifts the characters in a string, which is used to create the Vigenere matrix in `_create_matrix`. The matrix is a 2D list where each row represents the alphabet shifted by a certain number of positions.

3. Key Extension and Index Conversion

_extend_key ensures that the key is extended/repeated to match the length of the input string. _get_indexes converts characters into their corresponding index in the alphabet, which is used for matrix lookup during encryption/decryption.

4. Input Validation

`_validate_input` checks whether the input string only contains valid characters from the specified alphabet, ensuring data integrity.

5. Encryption and Decryption

encrypt_decrypt performs the actual encryption or decryption based on the operation argument. It validates the input, extends the key, converts characters to indexes, and performs the cipher operation using the Vigenere matrix.

Usage

When the script is executed, the menu function is called, providing the user with options to encrypt or decrypt messages using the Vigenere Cipher.

```
if __name__ == '__main__':
    menu()
```

Encryption Method

In the provided code, the encryption method is embedded within the **`encrypt_decrypt`** method, which is capable of handling both encryption and decryption based on an **operation** parameter. Let's break down the encryption process:

1. Preprocessing the Input

```
key = key.replace(' ', '').upper() string = string.replace(' ',
'').upper()
```

- **Remove Spaces**: Spaces are removed from the key and the input string to ensure a continuous block of text.
- **Uppercase Conversion**: Both the key and the input string are converted to uppercase to maintain consistency and simplify the encryption process.

2. Input Validation

```
if not (self._validate_input(key) and self._validate_input(string)):
print('Invalid input. Use letters from the specified alphabet only.')
return
```

- **Character Validation**: The **`_validate_input`** method checks whether every character in the key and string is present in the specified alphabet. If any invalid character is found, an error message is displayed, and the method returns without performing encryption.

3. Key Extension

```
extended_key = self._extend_key(string, key)
```

- **Matching Length**: The **`_extend_key`** method ensures that the key is extended (or repeated) to match the length of the input string. This is crucial because, in the Vigenere Cipher, each character in the plaintext is shifted according to a corresponding character in the key.

4. Index Conversion

```
key_indexes = self._get_indexes(extended_key) string_indexes =
self._get_indexes(string)
```

- **Alphabet Index**: The **`_get_indexes`** method converts characters into their corresponding index in the alphabet. This is done for both the extended key and the input string, facilitating the lookup operation in the Vigenere matrix during encryption.

5. Encryption Process

```
if operation == 'encrypt': return ''.join(self.matrix[k][v] for k, v in
zip(key_indexes, string_indexes))
```

- **Matrix Lookup**: For each character in the input string, the method uses the corresponding key character (both represented by their indexes) to find the encrypted character in the Vigenere matrix. Specifically, it uses the key index to select a row in the matrix and the string index to select a column, resulting in the ciphertext character.
- **Concatenation**: The characters are concatenated to form the final encrypted message (ciphertext).

Conclusion

The Vigenere Cipher was successfully implemented in Python, considering the specific constraints and requirements of handling the Romanian alphabet, ensuring valid character usage, and providing user interaction for encryption and decryption. The implementation adheres to clean coding principles, ensuring modularity, reusability, and maintainability of the code.

References

- Vigenere Cipher - Wikipedia
- GitHub Code Reference