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: Polialphabetical Chiper

Elaborated:
st.gr. FAF-211 Coreţchi Mihai
Verified:
asist.univ. Cătălin Mîţu

Chișinău, 2023

Playfair algorithm

Although it bears the name of Baron Lyon Playfair, the algorithm was invented by his friend, Charles Wheatstone, and first described in a document on March 26, 1854. Initially, it was rejected by the British Foreign Office because it was considered too difficult to understand. When Wheatstone offered to demonstrate that he could teach the algorithm to 3 out of 4 boys from a nearby school in 15 minutes, the Foreign Office secretary replied, 'Yes, it is very possible, but you will not be able to teach them to be good diplomats.' After the creation of the algorithm, Baron Playfair convinced the British government to adopt it for official use, and that's why it bears his name and not that of the creator, Wheatstone. The algorithm was used by the British army in the Boer War in South Africa, and modified versions were used by the British in World War I and by the Australian army in World War II. From the perspective of modern cryptography, the Playfair encryption algorithm is outdated, even primitive. Any modern personal computer can find (break) the key and decrypt the message in a matter of seconds or even fractions of a second, using the appropriate software. Some of the most skilled cryptanalysts or even some crossword experts can break the encrypted message in a few minutes using just a pencil and a piece of paper. Although it is an outdated algorithm in all respects, the Playfair algorithm is one of the first algorithms that uses the modern principles of block ciphers. Studying this algorithm can offer you a better intuitive understanding of ...

To encrypt a message using the Playfair cipher, you first create a 5x5 grid known as the key table. This table is filled with letters from a keyword, omitting any duplicates, followed by the remaining letters of the alphabet. Typically, the letters 'I' and 'J' are combined into a single cell. Once the key table is ready, you prepare the plaintext by dividing it into digraphs or pairs of letters. If a pair has the same letter twice or if there's a lone letter at the end, you insert a filler letter, commonly 'X', to complete the pair. To encrypt each digraph, you locate the letters in the key table. If the letters are in the same row, you replace them with the letters immediately to their right, wrapping around to the start of the row if necessary. If they are in the same column, you replace them with the letters immediately below them, wrapping around to the top if needed. If the letters form a rectangle, you replace them with the letters in the opposite corners of the rectangle along the same row. Decryption is the reverse process of encryption. You use the same key table and look for the digraphs in the ciphertext. If the letters are in the same row, you replace them with the letters immediately to their left, wrapping around if needed. If they are in the same column, you replace them with the letters immediately above them, wrapping around if necessary. If the letters form a rectangle, you replace them with the letters in the opposite corners of the rectangle along the same row.

Implementation

Implement the Playfair algorithm in one of the programming languages for messages in Romanian (31 letters). The character values of the text must be 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 must not be less than 7. The user will be able to choose the operation—encryption or decryption—and will be able to enter the key, the message, or the cryptogram, and will obtain the cryptogram or the decrypted message. The final phase of adding new spaces, depending on the language used and the logic of the message, will be done manually.

Input Cleaning: The next step involves cleaning the input text and key by eliminating all characters not part of the Romanian alphabet. Additionally, the text is converted to uppercase, making the encryption and decryption processes case-insensitive and standardized.

Key Validation: Before proceeding with encryption or decryption, the program validates the length of the key. If the key is shorter than 7 characters, the user is notified and prompted to enter a new key that meets the minimum length requirement.

Matrix Creation: The program constructs a 6x6 matrix, which is central to the Playfair cipher algorithm. This matrix is populated first with unique characters from the user's key and then filled with the remaining characters from the Romanian alphabet.

Character Positioning: The program identifies the row and column indices of each character in the text within the 6x6 matrix. These positions are crucial for applying the Playfair cipher rules during the encryption and decryption processes.

Cipher Operations: For each pair of characters in the text, the program applies the Playfair cipher rules based on their positions in the matrix. The characters are either shifted or swapped according to these rules, resulting in the encrypted or decrypted text.

User Interface: The main loop serves as the user interface, allowing the user to choose between encryption, decryption, or exiting the program. The program ensures that the text and key are cleaned and validated before executing the chosen operation and then displays the result.

```
/Users/mihaicoretchi/repos/CS/CS/Lab3/venv/bin/python /
Choose operation: encrypt | decrypt | exit: encrypt
Enter the text: student utm
Enter the key: laborator
Encrypted text: ȘRWÂÂSAYLȘ
Choose operation: encrypt | decrypt | exit: decrypt
Enter the text: ȘRWÂÂSAYLȘ
Enter the key: laborator
Decrypted text: STUDENTUTM
Choose operation: encrypt | decrypt | exit:
```

Figure 1

Result of applying the algorithm

Conclusion

In this laboratory work, I delved into the intricacies of the Playfair cipher, a classical encryption technique that encrypts digraphs—pairs of two letters—instead of single characters. I began by understanding the importance of string manipulation, which led me to use a specialized library to clean the input text and key. This step was crucial for ensuring that only valid characters from the Romanian alphabet were used, and it also made the encryption and decryption processes case-insensitive. I then focused on key validation, ensuring that the key met a minimum length requirement of 7 characters. This was an essential security measure to make the cipher more robust. Following this, I implemented the core of the Playfair cipher: the 6x6 matrix. Populating this matrix correctly was vital, as it served as the basis for all subsequent encryption and decryption operations. Understanding the positioning of characters within this matrix was my next focus. The row and column indices of each character played a pivotal role in the cipher operations. I applied the Playfair cipher rules to shift or swap characters based on their matrix positions, achieving the final encrypted or decrypted text. Lastly, I implemented a user interface within a main loop, allowing for real-time interaction with the program. This interface provided options for encryption, decryption, and exiting the program, making it user-friendly and versatile. Overall, this lab work provided me with a comprehensive understanding of the Playfair cipher, from input validation to the actual encryption and decryption processes. It was a rewarding experience that not only enhanced my programming skills but also deepened my appreciation for the complexities of cryptography.

https://github.com/eamtcPROG/CS/tree/main/Lab3