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	Grama Alexandru
Verified:	
asist.univ.	Catalin Mitu

Chișinău, 2023

Content

Vigenère cipher

The encryption method known as the ”Vigenère cipher” was incorrectly attributed to Blaise de Vigenère in the 19th century and was actually first described by Giovan Battista Bellaso in his 1553 book. Vigenère created a similar yet stronger and different cipher in 1586.

On the other hand, the Vigenère cipher uses the same operations as the Caesar cipher. The Vigenère cipher also shifts letters, but unlike Caesar, it cannot be easily broken into 26 combinations. The Vigenère cipher uses multiple shifts. The key is not made of a single shift but several, being generated by some integers ki, where 0 ki 25, if we take the Latin alphabet with 26 letters as a reference. The encryption is done as follows: ci=(mi+ki)mod26.ci=(mi+ki)mod26. The key can be, for example, k=(5,20,17,10,20,13)k=(5,20,17,10,20,13) and would cause the shift of the first letter by 5, c1=m1+5c1=m1+5 (mod 26), the second one by 20, c2=m2+20c2=m2+20 (mod 26), and so on until the end of the key, and then from the beginning, again. The key is usually a word to make it easier to remember - the above key corresponds to the word ”hose”. The multiple shift method offers additional protection for two reasons: The first reason is that others do not know the length of the key;
The second reason is that the number of possible solutions increases with the size of the key; for example, for the key length equal to 5, the number of combinations that would be necessary for exhaustive search would be 265=11,881,376.265=11,881,376.

Decryption for the Vigenère cipher is similar to encryption. The difference is that the key is subtracted from the encrypted text, $mi = (ciki)mod26.mi = (ciki)mod26$.

To simplify the encryption process, the following table can be used, called Tabula Recta, which was used by Vigenère. Here all 26 ciphers are located horizontally, and each cipher corresponds to a certain letter in the key, represented in the column on the left of the table. The alphabet corresponding to the letters of the plaintext is located in the top row of the table. The encryption process is simple - it is necessary that, having the letter mimi from the message and the letter kiki from the key, to find the letter of the encrypted text cici, which is located at the intersection of the mimi row and the kiki column. In the example from the case mi=Mmi=M and ki=Hki=H is presented, and as a result, ci=Tci=T is obtained.

Playfair Cipher

Although it bears the name of Baron Lyon Playfair, the algorithm was invented by his friend, Charles Wheatstone, and was 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 three out of four boys from a nearby school in 15 minutes, the foreign office secretary replied, “Yes, that 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 this algorithm for official use, hence it bears his name and not the creator, Wheatstone's. The algorithm was used by the British army in the war against the Boers 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 (crack) the key and decrypt the message in a matter of seconds or even milliseconds, using the right software. Some of the most skilled cryptanalysts or even some crossword experts can crack the encrypted message in a few minutes using just a pencil and a piece of paper.

Even though 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 provide a better intuitive understanding of modern cryptography without using complex mathematical knowledge or number theory.

This first step involves writing all letters in uppercase, in pairs, without spaces and punctuation. All 'J' letters in the text will be replaced by 'I' (in the example below, there is no 'J' letter). For example, consider the message: m = (the example message provided). First, it becomes (the example message in uppercase). Then, after dividing it into pairs of two letters, we get (the example message in pairs). The next step in preparing the text for encryption is inserting a 'Q', 'X', or 'Z' (which are the least common letters in the English vocabulary) between every double couple of letters. For instance, the word "FREEDOM" becomes "FR EX ED OM". This rule was introduced for two reasons:

Implementation

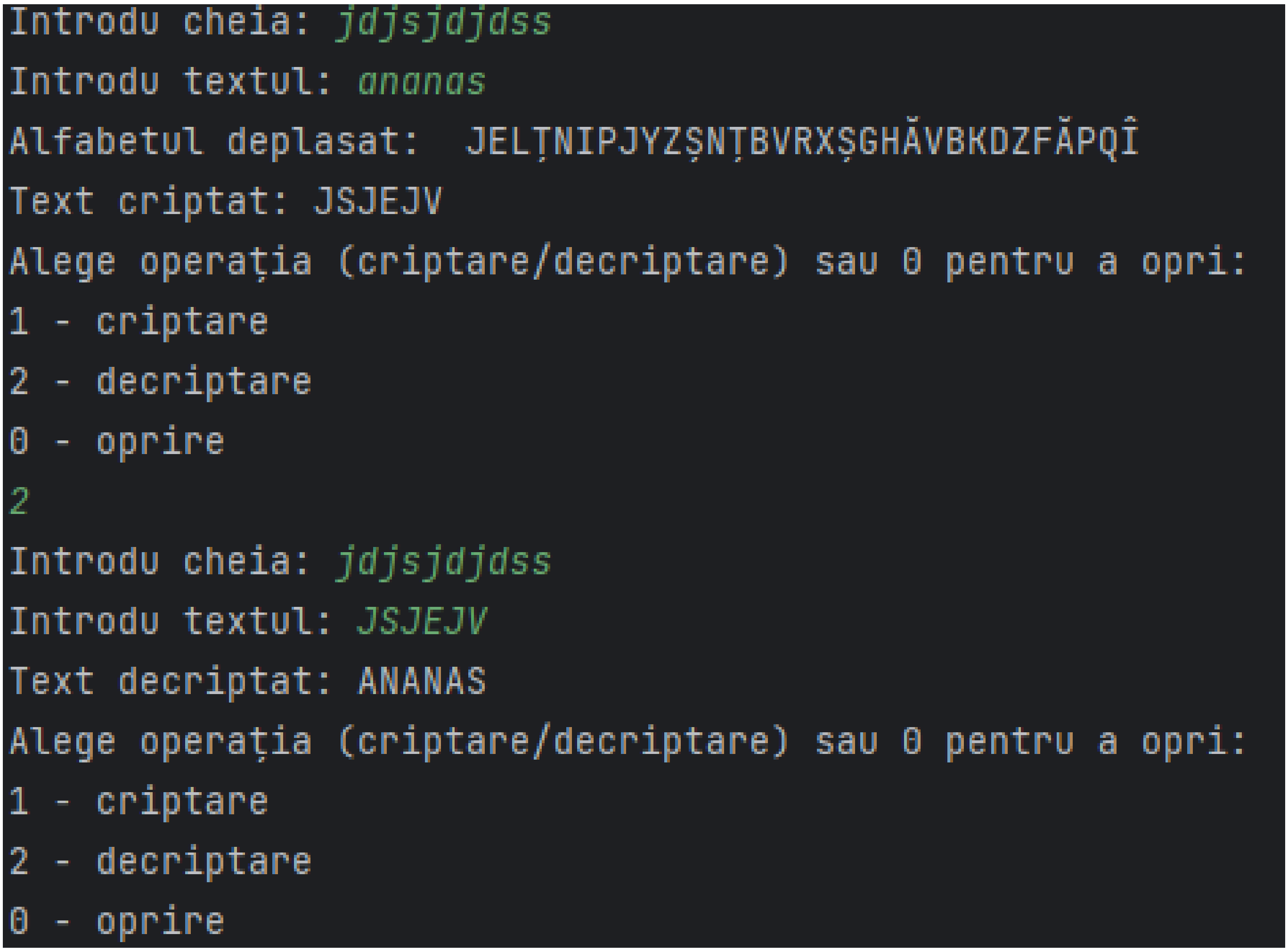
Task 3.2

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.

```
ROMAN_ALPHABET = ’AĂÂBCDEFGHIÎJKLMNOPQRSȘTȚUVWXYZ’


def validate_input(text):
    if not all(c.upper() in ROMAN_ALPHABET for c in text):
        raise ValueError("Textul poate conține doar litere din alfabetul român.")


def encode_char(c):
    if c.upper() in ROMAN_ALPHABET:
        return ROMAN_ALPHABET.index(c.upper())
    else:
        raise ValueError(f"Caracterul {c} nu este valid.")


def decode_char(c):
    return ROMAN_ALPHABET[c]


def vigenere_encrypt(plain_text, key):
    validate_input(plain_text)
    validate_input(key)

    if len(key) < 7:
        raise ValueError("Lungimea cheii trebuie să fie cel puțin 7.")
    plain_text = plain_text.replace(" ", "").upper()
    key = key.upper()

    encrypted_text = ""

    for i, c in enumerate(plain_text):
        ki = encode_char(key[i % len(key)])
        ci = (encode_char(c) + ki) % len(ROMAN_ALPHABET)
        encrypted_text += decode_char(ci)

    shifted_alphabet = shift_alphabet_based_on_key(key)
    print("Alfabetul deplasat: ", shifted_alphabet)
    return encrypted_text

def shift_alphabet_based_on_key(key):
    shifted_alphabet = ""
    key = key.upper()
    for i, c in enumerate(ROMAN_ALPHABET):
        ki = encode_char(key[i % len(key)])
        shifted_char = decode_char((encode_char(c) + ki) % len(ROMAN_ALPHABET))
        shifted_alphabet += shifted_char
    return shifted_alphabet

def vigenere_decrypt(cipher_text, key):
    validate_input(cipher_text)
    validate_input(key)

    if len(key) < 7:
        raise ValueError("Lungimea cheii trebuie să fie cel puțin 7.")

    cipher_text = cipher_text.replace(" ", "").upper()
    key = key.upper()

    decrypted_text = ""
    for i, c in enumerate(cipher_text):
        ki = encode_char(key[i % len(key)])
        ci = (encode_char(c) - ki) % len(ROMAN_ALPHABET)
        decrypted_text += decode_char(ci)

    return decrypted_text

while True:
    try:
        operation = input("Alege operația (criptare/decriptare) sau 0 pentru a opri:\n"
                          "1 - criptare\n"
                          "2 - decriptare\n"
                          "0 - oprire\n").lower()

        if operation == ’0’:
            print("Oprire program.")
            break

        key = input("Introdu cheia: ")
        text = input("Introdu textul: ")

        if operation == ’1’:
            print("Text criptat:", vigenere_encrypt(text, key))
        elif operation == ’2’:
            print("Text decriptat:", vigenere_decrypt(text, key))
        else:
            print("Operație invalidă.")
    except ValueError as e:
        print(e)
```

Screenshot:

Task 1.1



```
Introdu cheia: jdjsjdjdss
Introdu textul: ananas
Alfabetul deplasat:  JELȚNIPJYZȘNȚBVRXȘGHĂVBKDZFĂPQÎ
Text criptat: JSJEJV
Alege operația (criptare/decriptare) sau 0 pentru a opri:
1 - criptare
2 - decriptare
0 - oprire
2
Introdu cheia: jdjsjdjdss
Introdu textul: JSJEJV
Text decriptat: ANANAS
Alege operația (criptare/decriptare) sau 0 pentru a opri:
1 - criptare
2 - decriptare
0 - oprire
```

Conclusion

In the intricate world of cryptography, historical encryption methods such as the Vigenère cipher and Playfair algorithm have laid foundational groundwork that paved the way for the evolution of modern cryptographic practices. Though these methods are no longer considered secure or practical in the face of contemporary computational capabilities, their ingenuity and effectiveness during their time of inception are undeniable. Vigenère Cipher:

The Vigenère cipher, although incorrectly attributed to Blaise de Vigenère, showcased an enhanced level of security over the Caesar cipher by employing multiple shifts, effectively adding complexity and resistance against frequency analysis. This method was a significant leap towards creating more secure communication channels, yet it is rendered almost obsolete today given the computational prowess of modern computers and advanced cryptographic techniques. Playfair Algorithm:

Similarly, the Playfair algorithm, though named after Baron Lyon Playfair, was a creation of Charles Wheatstone. It introduced a new complexity by encrypting pairs of letters, marking a departure from the then-conventional single-letter encryption techniques. This method was crucial during wars and was instrumental in secure communications, marking its significance in cryptographic history. Relevance Today:

Though outdated, studying these historical encryption methods offers valuable insights into the evolution of cryptographic techniques. Both the Vigenère cipher and the Playfair algorithm underscore the unending race between encryption and decryption, the coded and the code-breakers. Their ingenious methods of securing communication highlight humanity's relentless pursuit of privacy and security.

In today's world of quantum computing and highly sophisticated cryptographic methods, historical ciphers like Vigenère and Playfair may appear rudimentary. However, they are testimonial to the constantly evolving nature of cryptography. They remind us of the necessity for adaptive, robust, and innovative solutions to stay ahead in the perpetual game of securing confidential information and communications. Every modern algorithm and cryptographic technique is built upon the lessons, failures, and successes of these pioneering methods.

In essence, while the Vigenère and Playfair ciphers are not utilized for secure communications in the modern context, they serve as vital educational tools for understanding the foundational principles of cryptography and the progression of encrypted communication over the centuries. They are stepping stones that have contributed to the intricate, multi-faceted world of contemporary cryptography, cyber security, and secure communications.