

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
 «ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**(ДГТУ)**

Факультет «Информатика и вычислительная техника»

Кафедра «Кибербезопасность информационных систем»

**Отчет**

**Лабораторная работа №4 по дисциплине «Методы и средства криптографической защиты»**

Выполнил:   
обучающийся гр. ВКБ41

Ушаков М.А.

Проверила:   
 Сафарьян Ольга Александровна

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ростов-на-Дону

2024

**Лабораторная работа №4**

**Цель работы:** формирование умений шифрования с использованием системы Вижинера и шифра «двойной квадрат» Уитстона.

Задание 1

Используя систему Вижинера, зашифруйте сообщения. Текст сообщения и ключевое слово должны соответствовать варианту задания лабораторной работы 3.

Вариант 24

Ключ.слово – ПАПКА

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| В | С | И | С | Т | Е | М | А | Х | П | Р | О | З | Р | А | Ч | Н | О | Г | О |
| П | А | П | К | А | П | А | П | К | А | П | А | П | К | А | П | А | П | К | А |
| 15 | 0 | 15 | 15 | 0 | 15 | 0 | 15 | 10 | 0 | 15 | 0 | 15 | 10 | 0 | 15 | 0 | 15 | 10 | 0 |
| С | С | Ч | Ы | Т | Ф | М | П | Я | П | Я | О | Ц | Ъ | А | Ж | Н | Э | Н | О |

Текст - В системах прозрачного шифрования преобразования осуществляются незаметно для пользователя

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ш | И | Ф | Р | О | В | А | Н | И | Я | П | Р | Е | О | Б | Р |
| П | А | П | К | А | П | А | П | К | А | П | А | П | К | А | П |
| 15 | 0 | 15 | 10 | 0 | 15 | 0 | 15 | 10 | 0 | 15 | 0 | 15 | 10 | 0 | 15 |
| З | И | Г | Ъ | О | С | А | Ь | Т | Я | Ю | Р | Ф | Ш | Б | Я |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| А | З | О | В | А | Н | И | Я | О | С | У | Щ | Е | С | Т | В | Л | Я | Ю | Т | С | Я |
| А | П | К | А | П | А | П | К | А | П | А | П | К | А | П | А | П | К | А | П | А | П |
| 0 | 15 | 10 | 0 | 15 | 0 | 15 | 10 | 0 | 15 | 0 | 15 | 10 | 0 | 15 | 0 | 15 | 10 | 0 | 15 | 0 | 15 |
| А | Ц | Ш | В | П | Н | Ч | Й | О | А | У | И | П | С | Б | В | Ъ | Й | Ю | Б | С | О |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Н | Е | З | А | М | Е | Т | Н | О | Д | Л | Я | П | О | Л | Ь | З | О | В | А | Т | Е | Л | Я |
| К | А | П | А | П | К | А | П | А | П | К | А | П | А | П | К | А | П | А | П | К | А | П | А |
| 10 | 0 | 15 | 0 | 15 | 10 | 0 | 15 | 0 | 15 | 10 | 0 | 15 | 0 | 15 | 10 | 0 | 15 | 0 | 15 | 10 | 0 | 15 | 0 |
| Ч | Е | Ц | А | Ы | П | Т | Ь | О | У | Х | Я | Ю | О | Ъ | Ж | З | Э | В | П | Ь | Е | Ъ | Я |

**ШИФР.ТЕКСТ:** – ССЧЫТФМПЯПЯОЦЪАЖНЭНОЗИГЪОСАЬТЯЮРФШБЯАЦШВПНЧЙОАУИПСБВЪЙЮБСОЧЕЦАЫПТЬОУХЯЮОЪЖЗЭВПЬЕЪЯ

Листинг программы:

def encrypt(text, key):

encrypted\_text = []

key\_index = 0

for char in text:

if char.upper() in alphabet:

shift = alphabet.index(key[key\_index]) % len(alphabet)

encrypted\_char = alphabet[(alphabet.index(char.upper()) + shift) % len(alphabet)]

encrypted\_text.append(encrypted\_char)

key\_index = (key\_index + 1) % len(key)

else:

encrypted\_text.append(char)

return ''.join(encrypted\_text)

def decrypt(text, key):

decrypted\_text = []

key\_index = 0

for char in text:

if char.upper() in alphabet:

shift = alphabet.index(key[key\_index]) % len(alphabet)

decrypted\_char = alphabet[(alphabet.index(char.upper()) - shift) % len(alphabet)]

decrypted\_text.append(decrypted\_char)

key\_index = (key\_index + 1) % len(key)

else:

decrypted\_text.append(char)

return ''.join(decrypted\_text)

#папка

#всистемахпрозрачногошифрованияпреобразованияосуществляютсянезаметнодляпользователя

alphabet = "АБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ"

text = input("Введите текст ").upper()

key = input("Введите ключ.слово ").upper()

encrypted = encrypt(text, key)

decrypted = decrypt(encrypted, key)

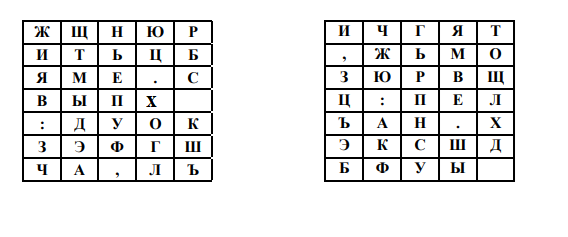
print("Зашифрованный текст:", encrypted)

print("Расшифрованный текст:", decrypted)

Задание 2

Используя шифр «двойной квадрат» Уитстона и шифрующие таблицы, представленные на рис. 4.1, выполните шифрование сообщения из задания лабораторной работы 3.

Текст - В системах прозрачного шифрования преобразования осуществляются незаметно для пользователя



**Шифр.текст** –

ЛЧЗРЩРВЬ ДЛПТБСЯФЩТЬДЦЕШЖЧТБ::ГЖЩЧПЕЩЬЬСБМ..УДМЖЛБРЪЯЫЩРЦ ЗМЬЭЩЧЯПК:ВЫЬДХЛХЫЩЧЛЬУЦДИ::МЫЫЮ

Листинг программы:

import numpy as np

def create\_matrix():

cyrillic\_chars = [

'А', 'Б', 'В', 'Г', 'Д', 'Е', 'Ж', 'З', 'И', 'К', 'Л', 'М', 'Н',

'О', 'П', 'Р', 'С', 'Т', 'У', 'Ф', 'Х', 'Ц', 'Ч', 'Ш', 'Щ', 'Ъ',

'Ы', 'Ь', 'Э', 'Ю', 'Я', ",", ".", " ", ":"

]

np.random.shuffle(cyrillic\_chars)

matrix = np.array(cyrillic\_chars).reshape(7, 5)

np.random.shuffle(cyrillic\_chars)

matrix\_sec = np.array(cyrillic\_chars).reshape(7, 5)

print(matrix)

print(matrix\_sec)

return matrix, matrix\_sec

def find\_position(square, char):

for i in range(7):

for j in range(5):

if square[i][j] == char:

return i, j

return None

def encrypt(plaintext, matrix, matrix\_sec):

encrypted\_text = ''

for i in range(0, len(plaintext), 2):

char1 = plaintext[i]

char2 = plaintext[i + 1]

row1, col1 = find\_position(matrix, char1)

row2, col2 = find\_position(matrix\_sec, char2)

if row1 == row2: # Одна строка

encrypted\_text += matrix\_sec[row1][col1]

encrypted\_text += matrix[row2][col2]

elif col1 == col2: # Один столбец

encrypted\_text += matrix\_sec[row1][col1]

encrypted\_text += matrix[row2][col2]

else:

encrypted\_text += matrix\_sec[row1][col2]

encrypted\_text += matrix[row2][col1]

return encrypted\_text

def decrypt(ciphertext, matrix, matrix\_sec):

decrypted\_text = ''

for i in range(0, len(ciphertext), 2):

char1 = ciphertext[i]

char2 = ciphertext[i + 1]

row1, col1 = find\_position(matrix\_sec, char1)

row2, col2 = find\_position(matrix, char2)

if row1 == row2: # Одна строка

decrypted\_text += matrix[row1][col1]

decrypted\_text += matrix\_sec[row2][col2]

elif col1 == col2: # Один столбец

decrypted\_text += matrix[row1][col1]

decrypted\_text += matrix\_sec[row2][col2]

else:

decrypted\_text += matrix[row1][col2]

decrypted\_text += matrix\_sec[row2][col1]

return decrypted\_text

text = input("Введите предложение ").upper()

if len(text) % 2 != 0:

text += "Ъ"

print(text)

example\_one = [

'Ж', 'Щ', 'Н', 'Ю', 'Р', 'И', 'Т', 'Ь', 'Ц', 'Б', 'Я', 'М', 'Е',

'.', 'С', 'В', 'Ы', 'П', 'Х', ' ', ':', 'Д', 'У', 'О', 'К', 'З',

'Э', 'Ф', 'Г', 'Ш', 'Ч', "А", ",", "Л", "Ъ"

]

example\_two = [

'И', 'Ч', 'Г', 'Я', 'Т', ',', 'Ж', 'Ь', 'М', 'О', 'З', 'Ю', 'Р',

'В', 'Щ', 'Ц', ':', 'П', 'Е', 'Л', 'Ъ', 'А', 'Н', '.', 'Х', 'Э',

'К', 'С', 'Ш', 'Д', 'Б', "Ф", "У", "Ы", " "

]

matrix = np.array(example\_one).reshape(7, 5)

matrix\_sec = np.array(example\_two).reshape(7, 5)

print(matrix,"\n",matrix\_sec)

encrypted\_text = encrypt(text, matrix, matrix\_sec)

# В системах прозрачного шифрования преобразования осуществляются незаметно для пользователя

print(f"Зашифрованный текст: {encrypted\_text}")

decrypted\_text = decrypt(encrypted\_text, matrix, matrix\_sec)

print(f"Расшифрованный текст: {decrypted\_text}")

**Вывод:** в ходе лабораторной работы было освоены и реализованы на языке программирования Pyhton алгоритм шифрования системы Вижинера и алгоритм шифра «двойной квадрат» Уитстона.