

# Математические основы защиты информации и информационной безопасности. Отчет по лабораторной работе № 2

## Шифры перестановка

Мохамед Либан Абдуллахи

### Содержание

Цель работы .....	1
Задание .....	1
Выполнение маршрутного шифрования.....	1
Выполнение шифрование с помощью решеток.....	3
Выполнение таблица виженера .....	4
Выводы .....	6
Список литературы .....	6

### Цель работы

Освоить на практике применение Шифры перестановка используя методы маршрутного шифрования, шифрование с помощью решеток и таблица виженера.

### Задание

- 1 маршрутное шифрование.
- 2 шифрование с помощью решеток.
- 3 таблица виженера.

### Выполнение маршрутного шифрования

Для выполнения работы была написана программа (@fig:1 - @fig:3) с помощью языка программирования Python, в исходном коде есть две функции, функция `groupingtextby_m()`, получает две параметра в качестве параметра исходный текст и ключ, и группирует исходного текст используя ключа, затем функция `encryption` принимает сгруппированный текст и шифрует методом маршрутного шифрования.

## Программа (1) исходный код

```
File Edit Selection View Go Run ... encryption algorithms
route_encryption.py X Vigenère_Cipher.py
route_encryption.py > encryption
1 punctuation=[" ", ",", ";", ":", "-", "!", "?", "\\", "'", "\""]
2
3 def groupingtextby_m(text, key):
4     m = len(key)
5     text = text.lower()
6     for punc in punctuation:
7         text = text.replace(punc, '')
8
9     text=[text[i:i+m] for i in range(0, len(text), m)]
10
11     if len(text[-1]) < m:
12         last_word = text[-1]
13         for i in range(0, m - len(last_word)):
14             last_word += ' '
15
16     text[-1] = last_word
17
18     return text
19
20 def encryption(grouped_text, key):
21     encrypted_text = ''
22     key_array = []
23     for chr in key:
24         key_array.append(ord(chr) - ord(' '))
25
26     for column in range(0, len(key_array)):
27
28         encrypted_word = ''
29
30         min_ele = min(key_array)
```

## Программа (2) исходный код

```
File Edit Selection View Go Run ... encryption algorithms
route_encryption.py X Vigenère_Cipher.py
route_encryption.py > encryption
31 min_ele = min(key_array)
32
33 i = key_array.index(min_ele)
34
35 for row in grouped_text:
36
37     j = grouped_text.index(row)
38     encrypted_word += grouped_text[j][i]
39
40     key_array[i] = 12345
41     encrypted_text += encrypted_word
42
43 return encrypted_text
44
45 key = "пароль"
46 plaintext = "нельзя недооценивать противника"
47 print("\n")
48 print("ORIGINAL TEXT: " + plaintext)
49 print("KEY: " + key)
50 print("\n")
51
52 grouped_text = groupingtextby_m(plaintext, key)
53
54 print("GROUPED TEXT BY KEY: ", grouped_text)
55
56 print("\n")
57 encrypted_text = encryption(grouped_text, key)
58 print("ENCRYPTED TEXT: " + encrypted_text)
```

## Вывод работы программы (3)

```
MINGW64/e/RUDN_Lessons/Semester1/Mathematical foundations of information protection and information security/Laps/work/2023-2024/MOZiiB/laboratory/lab02/encryption algorithms
Liban@DESKTOP-DV3061P MINGW64 /e/RUDN_Lessons/Semester1/Mathematical foundations of information protection and information security/Laps/work/2023-2024/MOZiiB/laboratory/lab02/encryption algo
rithms
$ ls
Vigenere_Cipher.py lattices_encryption.py route_encryption.py
Liban@DESKTOP-DV3061P MINGW64 /e/RUDN_Lessons/Semester1/Mathematical foundations of information protection and information security/Laps/work/2023-2024/MOZiiB/laboratory/lab02/encryption algo
rithms
$ python route_encryption.py

ORIGINAL TEXT: нельзя недооценивать противника
KEY: пароль

GROUPED TEXT BY KEY: ['нельзя', 'недооц', 'ениват', 'ьпроти', 'вникаа']

ENCRYPTED TEXT: еенпнзоатаьовонкнеьвдиряцтна
Liban@DESKTOP-DV3061P MINGW64 /e/RUDN_Lessons/Semester1/Mathematical foundations of information protection and information security/Laps/work/2023-2024/MOZiiB/laboratory/lab02/encryption algo
rithms
$ |
```

## Выполнение шифрование с помощью решеток

Для выполнения работы была написана программа (@fig:1 - @fig:2) с помощью языка программирования Python, в исходном коде есть две функции, Функция транспонирования берет таблицу размером 4x4, представленную в виде кортежа строк, и поворачивает ее на 90 градусов по часовой стрелке, перебирая столбцы и меняя местами символы в каждом столбце, функция recall\_password принимает два аргумента: таблицу 4x4 с заполнителями 'X' и таблицу кодов 4x4. и шифрует методом шифрование с помощью решеток.

Программа (1) исходный код

```
File Edit Selection View Go Run ... encryption algorithms
lattices_encryption.py X Vigenere_Cipher.py
lattices_encryption.py > recall_password
1 def transposition(tabl):
2     a = []
3     for i in range(4):
4         s = ''
5         for b in reversed(tabl):
6             s += b[i]
7         a.append(s)
8     return a
9
10 def recall_password(tabl, cod):
11     txt = ''
12     for k in range(4):
13         for i in range(4):
14             for j in range(4):
15                 if tabl[i][j] == "X":
16                     txt = txt + cod[i][j]
17             tabl=transposition(tabl)
18     return txt
19
20 print ( recall_password(
21     ('X...',
22      '..X.',
23      'X..X',
24      '....'),
25     ('itdf',
26      'gdce',
27      'aton',
28      'qrdi')))
```

Вывод работы программы (2)

```
MINGW64/e/RUDN_Lessons/Semester1/Mathematical foundations of information protection and information security/Laps/work/2023-2024/MOZiiB/laboratory/lab02/encryption algorithms
Liban@DESKTOP-DV3061P MINGW64 /e/RUDN_Lessons/Semester1/Mathematical foundations of information protection and information security/Laps/work/2023-2024/MOZiiB/laboratory/lab02/encryption algorithms
$ ls
Vigenère_Cipher.py  lattices_encryption.py  route_encryption.py
Liban@DESKTOP-DV3061P MINGW64 /e/RUDN_Lessons/Semester1/Mathematical foundations of information protection and information security/Laps/work/2023-2024/MOZiiB/laboratory/lab02/encryption algorithms
$ python lattices_encryption.py
icantforgetiddqd
Liban@DESKTOP-DV3061P MINGW64 /e/RUDN_Lessons/Semester1/Mathematical foundations of information protection and information security/Laps/work/2023-2024/MOZiiB/laboratory/lab02/encryption algorithms
$ |
```

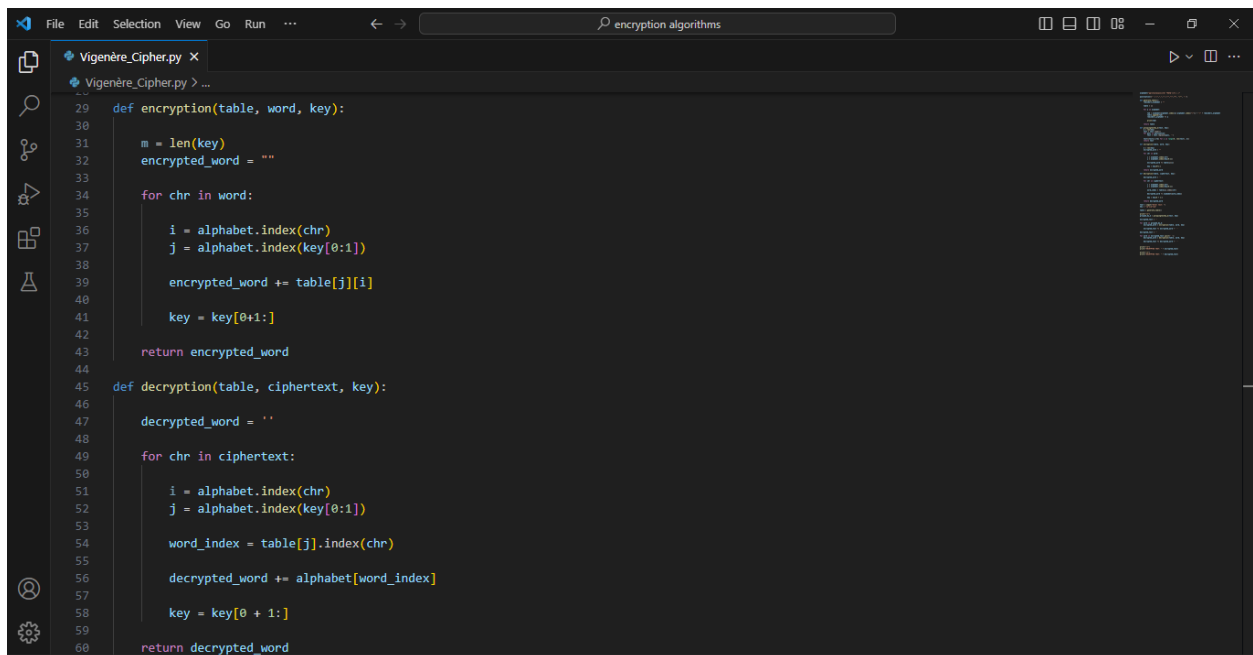
## Выполнение таблица виженера

Для выполнения работы была написана программа (@fig:1 - @fig:4) с помощью языка программирования Python, функция `generate_table()` генерирует таблицу используя алфавиты, `groupingtextby_m()` получает две параметри в качестве параметра исходный текст и ключ, и группирует исходного тескт используя ключа, затем функция `encryption` и `decryption` принимает таблицу, слова которое будем шифровать или расшифровать и ключ затем шифрует/расшифрует методом шифрование виженера.

Программа (1) исходный код

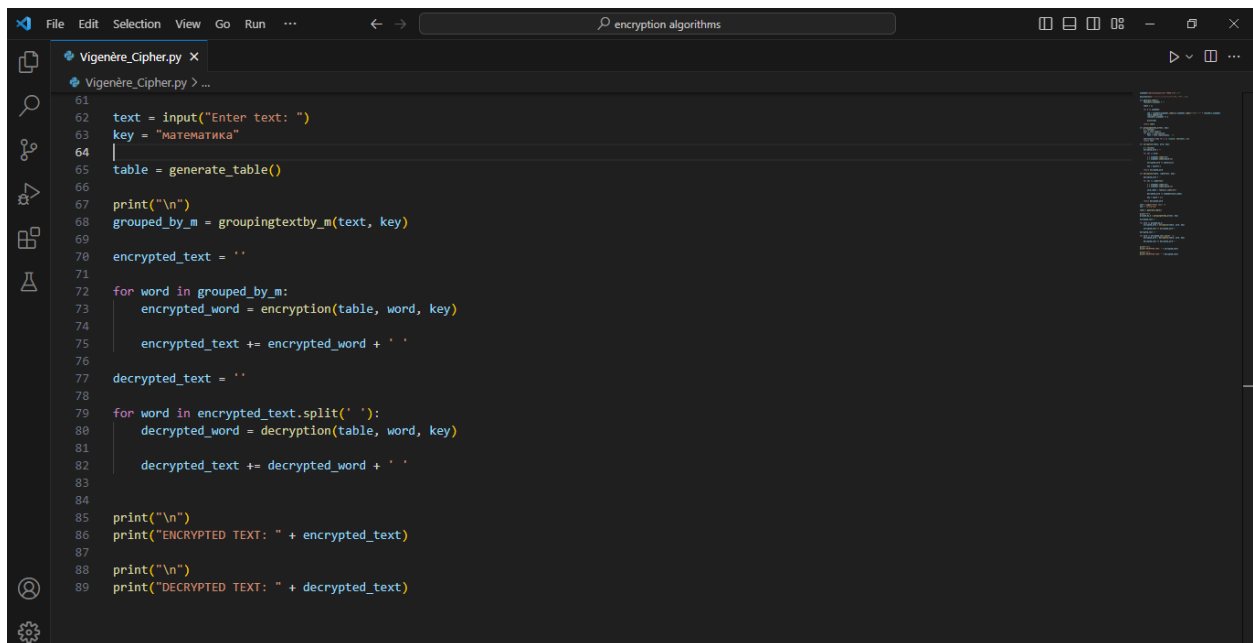
```
File Edit Selection View Go Run ... encryption algorithms
Vigenère_Cipher.py X
Vigenère_Cipher.py > ...
1 alphabet="абггдеёжзийклмнопрстуфхцчщъыьэя"
2
3 punctuation=[" ", ",", ".", ":", ";", "!", "?", "\n", "\t", "."]
4
5 def generate_table():
6     reminders_alphabet = ""
7
8     table = []
9
10    for a in alphabet:
11
12        tab = alphabet[alphabet.index(a):alphabet.index("а")] + "а" + reminders_alphabet
13        table.append(tab)
14        reminders_alphabet += a
15
16    print(table)
17
18    return table
19
20 def groupingtextby_m(text, key):
21     m = len(key)
22     text = text.lower()
23     for punc in punctuation:
24         text = text.replace(punc, '')
25
26     text=[text[i:i+m] for i in range(0, len(text), m)]
27     return text
28
```

Программа (2) исходный код



```
File Edit Selection View Go Run ... encryption algorithms
Vigenère_Cipher.py X
Vigenère_Cipher.py > ...
29 def encryption(table, word, key):
30
31     m = len(key)
32     encrypted_word = ""
33
34     for chr in word:
35
36         i = alphabet.index(chr)
37         j = alphabet.index(key[0:1])
38
39         encrypted_word += table[j][i]
40
41         key = key[0+1:]
42
43     return encrypted_word
44
45 def decryption(table, ciphertext, key):
46
47     decrypted_word = ""
48
49     for chr in ciphertext:
50
51         i = alphabet.index(chr)
52         j = alphabet.index(key[0:1])
53
54         word_index = table[j].index(chr)
55
56         decrypted_word += alphabet[word_index]
57
58         key = key[0 + 1:]
59
60     return decrypted_word
```

Программа (3) исходный код



```
File Edit Selection View Go Run ... encryption algorithms
Vigenère_Cipher.py X
Vigenère_Cipher.py > ...
61
62 text = input("Enter text: ")
63 key = "математика"
64
65 table = generate_table()
66
67 print("\n")
68 grouped_by_m = groupingtextby_m(text, key)
69
70 encrypted_text = ""
71
72 for word in grouped_by_m:
73     encrypted_word = encryption(table, word, key)
74     encrypted_text += encrypted_word + " "
75
76 decrypted_text = ""
77
78 for word in encrypted_text.split(' '):
79     decrypted_word = decryption(table, word, key)
80     decrypted_text += decrypted_word + " "
81
82 print("\n")
83 print("ENCRYPTED TEXT: " + encrypted_text)
84
85 print("\n")
86 print("DECRYPTED TEXT: " + decrypted_text)
```

Вывод работы программы (4)

