

# **Отчет по лабораторной работе №8**

**Лабораторная работа №8: Элементы криптографии. Шифрование (кодирование) различных исходных текстов одним ключом.**

Ширяев Кирилл Владимирович, НФИбд-03-18

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Выполнение работы</b>	<b>5</b>
2.1	Алфавит .....	4
2.2	Сообщения .....	4
2.3	Ключ .....	5
2.4	Перевод сообщений.....	6
2.5	Шифрование .....	7
2.6	Способ, прочтения одного из открытых текстов .....	9
2.7	Проверка .....	10
<b>3</b>	<b>Выводы</b> .....	<b>11</b>

## List of Figures

2.1	Алфавиты .....	5
2.2	Сообщения .....	6
2.3	Создание ключа .....	7
2.4	Шестнадцетиричная система .....	8
2.5	Шестнадцетиричная система .....	8
2.6	Шифрование .....	9
2.7	Шифрование .....	10
2.8	Прочтение .....	11
2.9	Проверка .....	11

## 1 Цель работы

Освоить на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом.

## 2 Выполнение работы

### 2.1 Алфавит

Задала алфавит из русских буквы алфавит из соответствующих им шестнадцатеричных чисел.

```
1 "к", "л", "м", "н", "о", "п", "р", "с", "т", "у", "ф", "х", "ц", "ч", "ш", "щ", "ъ", "ы", "ь", "э", "ю", "я", " ", "[", "]",  
2  
3 alph_16=[]  
4 q=hex(int('c0', 16))  
5 for i in range(64):  
6     alph_16.append(q)  
7     q=hex(int(q, 16)+int('1',16))  
8     alph_16.append(hex(int('20',16)))  
9     alph_16.append(hex(int('21',16)))  
10    alph_16.append(hex(int('22',16)))
```

Figure 2.1: Алфавиты

### 2.2 Сообщения

Ввел сообщения.

```
1 line_1 = 'С Новым Годом, друзья!'  
2 len(line_1)
```

22

```
1 line_2 = 'С Новым Мячом, друзья!'  
2 len(line_2)
```

22

```
1 list_1=list(line_1)
```

```
1 list_2=list(line_2)
```

Figure 2.2: Сообщения

## 2.3 Ключ

Создал рандомный ключ.

```
1 from random import randint
2
3 key=[]
4 for i in range(len(line_1)):
5     x=randint (0,255)
6     x=hex(x)
7     key.append(x)
8     print(x.replace("0x",""))
```

c0  
6f  
df  
89  
58  
36  
16  
cd  
af  
bb  
5b  
54  
94  
13  
16  
17  
4a  
ff  
58  
5e  
42  
b

Figure 2.3: Создание ключа

## 2.4 Перевод сообщений

Перевел заданные сообщение в шестнадцетиричные числа.

```

1 list_16_1=[]
2 def into_list_16(list_1, alphabet, alphabet_16, list_16):
3     for i in range(len(list_1)):
4         for j in range(len(alphabet)):
5             if list_1[i]==alphabet[j]:
6                 for k in range (len(alphabet_16)):
7                     if j==k:
8                         list_16.append(alphabet_16[k])
9                         print(alphabet_16[k].replace("0x", ""))
10 into_list_16(list_1, alph, alph_16, list_16_1)

```

```

d1
20
cd
ee
e2
fb
ec
20
c3
ee
e4
ee
ec
22
20
e4
f0
f3
e7
fc
ff
21

```

Figure 2.4: Шестнадцетиричная система

```

1 list_16_2=[]
2 def into_list_16(list_1, alphabet, alphabet_16, list_16):
3     for i in range(len(list_1)):
4         for j in range(len(alphabet)):
5             if list_1[i]==alphabet[j]:
6                 for k in range (len(alphabet_16)):
7                     if j==k:
8                         list_16.append(alphabet_16[k])
9                         print(alphabet_16[k].replace("0x", ""))
10 into_list_16(list_2, alph, alph_16, list_16_2)

```

```

d1
20
cd
ee
e2
fb
ec
20
cc
ff
f7
ee
ec
22
20
e4
f0
f3
e7
fc
ff
21

```

Figure 2.5: Шестнадцетиричная система

## 2.5 Шифрование

Зашифровал два сообщения с помощью одного и того же ключа.

```

1 cipher_1=[]
2 def into_cipher(list_16, key, cipher):
3     for i in range(len(list_16)):
4         for j in range(len(key)):
5             if i==j:
6                 x=hex(int(list_16[i],16) ^ int(key[j],16))
7                 cipher.append(x)
8                 print(x.replace("0x", ""))
9 into_cipher(list_16_1, key, cipher_1)

```

11  
4f  
12  
67  
ba  
cd  
fa  
ed  
6c  
55  
bf  
ba  
78  
31  
36  
f3  
ba  
c  
bf  
a2  
bd  
2a

Figure 2.6: Шифрование



```

1 cipher_2=[]
2 def into_cipher(list_16, key, cipher):
3     for i in range(len(list_16)):
4         for j in range(len(key)):
5             if i==j:
6                 x=hex(int(list_16[i],16) ^ int(key[j],16))
7                 cipher.append(x)
8                 print(x.replace("0x",""))
9 into_cipher(list_16_2, key, cipher_2)

```

11  
4f  
12  
67  
ba  
cd  
fa  
ed  
63  
44  
ac  
ba  
78  
31  
36  
f3  
ba  
c  
bf  
a2  
bd  
2a

Figure 2.7: Шифрование

## 2.6 Способ, прочтения одного из открытых текстов

Способ, при котором злоумышленник может прочесть оба текста, не зная ключа, но зная один из открытых текстов и два зашифровки текстов.

```

1 P1=[]
2 def get_P(P1, P2, C1, C2):
3     for i in range(len(C1)):
4         for j in range(len(C2)):
5             if i==j:
6                 for k in range(len(P2)):
7                     if j==k:
8                         x=hex(int(C1[i],16) ^ int(C2[j],16))
9                         x.replace("0x","")
10                        x=hex(int(P2[k],16) ^ int(x,16))
11                        P1.append(x)
12                        print(x.replace("0x",""))
13 get_P(P1, list_16_2, cipher_1, cipher_2)

```

d1  
20  
cd  
ee  
e2  
fb  
ec  
20  
c3  
ee  
e4  
ee  
ec  
22  
20  
e4  
f0  
f3  
e7  
fc  
ff  
21

Figure 2.8: Прочтение

## 2.7 Проверка

Проверка

```

1 if list_16_1==P1:
2     print('Yes')

```

Yes

Figure 2.9: Проверка

## 3 Выводы

Я освоил на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом.