

# **Отчет по лабораторной работе №7**

**Лабораторная работа №7: Элементы криптографии. Однократное гаммирование.**

Голова Варвара Алексеевна, НФИбд-03-18

2021, 11 December

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Выполнение работы</b>	<b>5</b>
2.1	Алфавит . . . . .	5
2.2	Сообщение . . . . .	5
2.3	Ключ . . . . .	6
2.4	Перевод сообщения . . . . .	6
2.5	Шифрование . . . . .	7
2.6	Расшифровка . . . . .	8
2.7	Перевод сообщения . . . . .	8
2.8	Ключ . . . . .	9
<b>3</b>	<b>Выводы</b>	<b>11</b>

# List of Figures

2.1	Алфавиты . . . . .	5
2.2	Сообщение . . . . .	5
2.3	Создание ключа . . . . .	6
2.4	Шестнадцетиричная система . . . . .	7
2.5	Шифрование . . . . .	8
2.6	Расшифровка . . . . .	8
2.7	Шестнадцетиричная система . . . . .	9
2.8	Возможный ключ . . . . .	10

# **1 Цель работы**

Освоить на практике применение режима однократного гаммирования.

## 2 Выполнение работы

### 2.1 Алфавит

Задала алфавит из русских букв и алфавит из соответствующих им шестнадцатеричных чисел.

```
Ввод [1]: 1 alph=["А", "Б", "В", "Г", "Д", "Е", "Ж", "З", "И", "Й", "К", "Л", "М", "Н", "О", "П", "Р", "С", "Т", "У", "Ф", "Х", "Ц", "Ч"]
          2
          3
Ввод [2]: 1 alph_16=[]
          2 q=hex(int('с0', 16))
          3 for i in range(64):
          4     alph_16.append(q)
          5     q=hex(int(q, 16)+int('1', 16))
          6 alph_16.append(hex(int('20', 16)))
          7 alph_16.append(hex(int('21', 16)))
          8 alph_16.append(hex(int('22', 16)))
```

Figure 2.1: Алфавиты

### 2.2 Сообщение

Ввела сообщение.

```
Ввод [3]: 1 line = 'С Новым Годом, друзья!'
          2 len(line)
```

Out[3]: 22

```
Ввод [4]: 1 list_1=list(line)
```

Figure 2.2: Сообщение

## 2.3 Ключ

Создала случайный ключ.

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

e7
34
46
83
1
f
a9
22
b9
16
e8
5b
48
a0
48
f1
5a
25
8e
26
5c
de
```

Figure 2.3: Создание ключа

## 2.4 Перевод сообщения

Перевела заданное сообщение в шестнадцатеричные числа.

```

1 list_16=[]
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)

```

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

## 2.5 Шифрование

Определила вид шифротекста при известном ключе и известном открытом тексте.

```

Ввод [8]: 1 cipher=[]
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, key, cipher)

36
14
8b
6d
e3
f4
45
2
7a
f8
c
b5
a4
82
68
15
aa
d6
69
da
a3
ff

```

Figure 2.5: Шифрование

## 2.6 Расшифровка

Один из вариантов расшифровки полученного шифра.

```

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

```

22

```

1 list_2=list(line_1)
2 list_2

```

Figure 2.6: Расшифровка

## 2.7 Перевод сообщения

Перевела один из возможных вариантов расшифровки сообщения в шестнадцетиричные числа.



1	list_16_2=[]
2	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 2.7: Шестнадцетиричная система

## 2.8 Ключ

Определила ключ, с помощью которого шифротекст может быть преобразован в некоторый фрагмент текста, представляющий собой один из возможных вариантов прочтения открытого текста.

```

Ввод [12]: 1 key_2=[]
2 def into_key(cipher, list_16, key):
3     for i in range(len(cipher)):
4         for j in range(len(list_16)):
5             if i==j:
6                 x=hex(int(cipher[i],16) ^ int(list_16[j],16))
7                 key_2.append(x)
8                 print(x.replace("0x",""))
9 into_key(cipher, list_16_2, key_2)

e7
34
46
83
1
f
a9
22
b6
7
fb
5b
48
a0
48
f1
5a
25
8e
26
5c
de

```

Figure 2.8: Возможный ключ

## **3 Выводы**

Я освоила на практике применение режима однократного гаммирования.