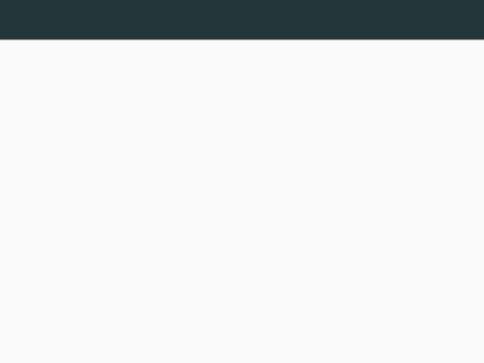
Лабораторная работа №8: Элементы криптографии. Шифрование (кодирование) различных исходных текстов одним ключом.

дисциплина: Информационная безопасность

Цель работы



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

Выполнение работы

Задал алфавит из русских букв и алфавит из соответствующих им шестнадцетиричных чисел.

Figure 1: Алфавиты

Ввел сообщения.

```
line 1 = 'C Новым Годом, друзья!'
 2 len(line_1)
22
   line 2 = 'C Новым Мячом, друзья!'
   len(line 2)
22
   list 1=list(line 1)
 1 list_2=list(line_2)
```

Figure 2: Сообщения

```
from random import randint
    key=[]
    for i in range(len(line_1)):
        x=randint (0,255)
        x=hex(x)
 6
 7
        key.append(x)
        print(x.replace("0x",""))
 8
cØ
6f
df
89
58
36
16
cd
af
bb
5b
54
94
13
16
17
4a
ff
58
```

Создал рандомный ключ.

Перевел заданные сообщение в шестнадцетиричные числа.

```
1 list_16_1=[]
 2 def into list 16(list 1, alphabet, alphabet 16, list 16):
        for i in range(len(list 1)):
            for j in range(len(alphabet)):
                if list_1[i] == alphabet[j]:
                    for k in range (len(alphabet 16)):
                        if j==k:
                            list_16.append(alphabet_16[k])
                            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
ee
e4
ee
ec
22
20
64
fø
e7
fc
ff
21
```

Figure 4: Шестнадцетиричная система

Перевел заданные сообщение в шестнадцетиричные числа.

```
1 list 16 2=[]
 2 def into list 16(list 1, alphabet, alphabet 16, list 16):
        for i in range(len(list_1)):
            for j in range(len(alphabet)):
                if list_1[i] == alphabet[j]:
                    for k in range (len(alphabet 16)):
                        if j==k:
                            list 16.append(alphabet 16[k])
                            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
fø
f3
e7
fc
ff
21
```

Figure 5: Шестнадцетиричная система

Зашифровал сообщения с помощью ключа.

```
1 cipher 1=[]
 2 def into_cipher(list_16, key, cipher):
        for i in range(len(list 16)):
 4
            for j in range(len(key)):
                if i==j:
                     x=hex(int(list_16[i],16) ^ int(key[j],16))
                     cipher.append(x)
                     print(x.replace("0x",""))
 8
   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
```

```
1 cipher 2=[]
   def into_cipher(list_16, key, cipher):
        for i in range(len(list_16)):
            for j in range(len(key)):
                if i== j:
                    x=hex(int(list_16[i],16) ^ int(key[j],16))
                    cipher.append(x)
                    print(x.replace("0x",""))
   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
```

0-

Зашифровал сообщения с помощью ключа.

```
1 P1=[]
 2 def get_P(P1, P2, C1, C2):
        for i in range(len(C1)):
            for j in range(len(C2)):
                if i==j:
                    for k in range(len(P2)):
                        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))
                            P1.append(x)
                            print(x.replace("0x",""))
12
13 get_P(P1, list_16_2, cipher_1, cipher_2)
d1
20
cd
ee
e2
fh
ec
20
c3
ee
e4
ee
ec
22
20
e4
fø
```

f3

Способ, при котором злоумышленник может прочитать оба текста, не зная ключа.

Проверка

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

Yes

Figure 9: Проверка

Выводы

Я освоил на практике применение режима однократного гаммирования на примере кодирования различных исходных

текстов одним ключом.