

Лабораторная работа №8

Основы Информационной Безопасности

Чистов Даниил Максимович

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Выводы	10
4	Список литературы	11

Список иллюстраций

2.1	Функции шифрования и дешифрования	5
2.2	Функция взлома	6
2.3	Результат взлома	7

1 Цель работы

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

2 Выполнение лабораторной работы

Требуется написать программу, которая сможет получить расшифрованное сообщение 2, зная только оба зашифрованных сообщения и расшифрованное сообщение 1, без наглядного поиска ключа - на первом фото: функции шифрования и дешифрования текстов. Принцип такой: у нас есть алфавит из русских больших, маленьких букв и цифр. У каждого символа есть индекс. Также индексы есть и у сообщений. Складываем индексы сообщения и ключа, так получаем новый индекс и достаём из алфавита другую букву, соответствующую новому индексу. Дешифрование реализуется также, но вместо сложения, происходит вычитание (рис. 2.1).

```
1 alphabet = [  
2     "а", "б", "в", "г", "д", "е", "ё", "ж", "з", "и", "й", "к", "л", "м", "н", "о",  
3     "п", "р", "с", "т", "у", "ф", "х", "ц", "ч", "ш", "щ", "ъ", "ы", "ь", "э", "ю", "я",  
4     "А", "Б", "В", "Г", "Д", "Е", "Ё", "Ж", "З", "И", "Й", "К", "Л", "М", "Н", "О",  
5     "П", "Р", "С", "Т", "У", "Ф", "Х", "Ц", "Ч", "Ш", "Щ", "Ъ", "Ы", "Ь", "Э", "Ю", "Я",  
6     "0", "1", "2", "3", "4", "5", "6", "7", "8", "9"  
7 ]  
8  
9 # Шифрование  
10 def encrypt(text, key): 5 usages  
11     res = ''  
12     for i in range(0, len(text)):  
13         i1 = alphabet.index(text[i])  
14         i2 = alphabet.index(key[i])  
15         res += alphabet[(i1 + i2) % len(alphabet)]  
16     return res  
17  
18 # Дешифрование  
19 def decrypt(text, key): 2 usages  
20     res = ''  
21     for i in range(0, len(text)):  
22         i1 = alphabet.index(text[i])  
23         i2 = alphabet.index(key[i])  
24         res += alphabet[(i1 - i2) % len(alphabet)]  
25     return res  
26  
27
```

Рис. 2.1: Функции шифрования и дешифрования

На втором фото: функция получения расшифрованного сообщения 2, зная сооб-

щение 1 и оба зашифрованных сообщений. Принцип такой, т.к. чтобы получить ключ, нам требовалось вычесть из индекса буквы сообщения индекс буквы ключа, то теперь, чтобы по сути получить ключ мы вычитаем из индекса буквы зашифрованного сообщения индекс буквы расшифрованного сообщения, так мы по сути получили индекс буквы ключа, который можем вычесть из зашифрованного сообщения (рис. 2.1).

```
30 # Проверка работы шифрования/дешифрования
31 print('1.1 - Проверка работы шифрования/дешифрования \n')
32 soobsh1 = 'ёлкиголкипомогитектонибудь'
33 klyuch1 = 'всемприветдорогиеизрителимо'
34
35 print('она, зашифровали \n', encrypt(soobsh1, klyuch1))
36
37
38 if soobsh1 == decrypt(encrypt(soobsh1, klyuch1), klyuch1):
39     print('AAA РАСШИФРОВАЛИ')
40     print(decrypt(encrypt(soobsh1, klyuch1), klyuch1))
41
42
43 print('\nРешение лабораторной работы')
44
45
46 def operation_vzломchik(zash1, zash2, izv1): 2 usages
47     res = ''
48     for i in range(0, len(zash1)):
49         c1 = alphabet.index(zash1[i])
50         c2 = alphabet.index(zash2[i])
51         p1 = alphabet.index(izv1[i])
52         res += alphabet[(c2 - (c1 - p1)) % len(alphabet)]
53
54     return res
55
```

Рис. 2.2: Функция взлома

На третьем фото: результат работы всей программы

```

s1 = 'здравствуйтеняязовутчистовданиил'
s2 = 'максимовичгруппанкабд0323изтопроц'
k1 = 'ессвыполнениялабораторнойработыно'

s1_unc = encrypt(s1, k1)
s2_unc = encrypt(s2, k1)

# print(operation_vzломchik(s1_unc, s2_unc, s1))

if s2 == operation_vzломchik(s1_unc, s2_unc, s1):
    print('Нас рассекретили :(')
    print('Неизвестное ранее сообщение:', s2)
    print('Сообщение, которые мы получили по средствам взлома:', operation_vzломchik(s1_unc, s2_unc, s1))

```

script x

C:\Users\12232\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\12232\PyCharmMiscProject\script.py

1.1 - Проверка работы шифрования/дешифрования

опа, зашифровали
 ээлхшучнпыуэээсчнтГчАнмьрК
 ААА РАСШИФРОВАЛИ
 ёлкииголкипомогитектонибудь

Решение лабораторной работы
 Нас рассекретили :(
 Неизвестное ранее сообщение: максимовичгруппанкабд0323изтопроц
 Сообщение, которые мы получили по средствам взлома: максимовичгруппанкабд0323изтопроц

Рис. 2.3: Результат взлома

Сам код:

```

alphabet = [
    "a", "б", "в", "г", "д", "е", "ё", "ж", "з", "и", "й", "к", "л", "м", "н", "о",
    "п", "р", "с", "т", "у", "ф", "х", "ц", "ч", "ш", "щ", "ъ", "ы", "ь", "э", "ю", "я",
    "А", "Б", "В", "Г", "Д", "Е", "Ё", "Ж", "З", "И", "Й", "К", "Л", "М", "Н", "О",
    "П", "Р", "С", "Т", "У", "Ф", "Х", "Ц", "Ч", "Ш", "Щ", "Ъ", "Ы", "Ь", "Э", "Ю", "Я",
    "0", "1", "2", "3", "4", "5", "6", "7", "8", "9"
]

```

Шифрование

```

def encrypt(text, key):
    res = ''
    for i in range(0, len(text)):
        i1 = alphabet.index(text[i])
        i2 = alphabet.index(key[i])
        res += alphabet[(i1 + i2) % len(alphabet)]

```

```

    return res

# Дешифрование
def decrypt(text, key):
    res = ''
    for i in range(0, len(text)):
        i1 = alphabet.index(text[i])
        i2 = alphabet.index(key[i])
        res += alphabet[(i1 - i2) % len(alphabet)]
    return res

# Проверка работы шифрования/дешифрования
print('1.1 - Проверка работы шифрования/дешифрования \n')
soobsh1 = 'ёлкииголкипомогитектонибудь'
klyuch1 = 'всемприветдорогиеизрителимо'

print('она, зашифровали \n', encrypt(soobsh1, klyuch1))

if soobsh1 == decrypt(encrypt(soobsh1, klyuch1), klyuch1):
    print('AAA РАСШИФРОВАЛИ')
    print(decrypt(encrypt(soobsh1, klyuch1), klyuch1))

print('\nРешение лабораторной работы')
```



```

def operation_vzломchik(zash1, zash2, izv1):
    res = ''
    for i in range(0, len(zash1)):
        c1 = alphabet.index(zash1[i])
        c2 = alphabet.index(zash2[i])
        p1 = alphabet.index(izv1[i])
        res += alphabet[(c2 - (c1 - p1)) % len(alphabet)]

    return res

s1 = 'здравствуйтеменязовутчистовданиил'
s2 = 'максимовичгруппанкабд0323изэтопроц'
k1 = 'ессыполнениялабораторнойработыно'

s1_unc = encrypt(s1, k1)
s2_unc = encrypt(s2, k1)

# print(operation_vzломchik(s1_unc, s2_unc, s1))

if s2 == operation_vzломchik(s1_unc, s2_unc, s1):
    print('Нас рассекретили :(')
    print('Неизвестное ранее сообщение:', s2)
    print('Сообщение, которые мы получили по средствам взлома:', operation_vzломchik(s

```

3 Выводы

В результате выполнения данной лабораторной работы я освоил на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом

4 Список литературы

Лабораторная работа №8