

# Лабораторная работа №7. Элементы криптографии. Однократное гаммирование

Захарова Софья Михайловна

## Цель работы

Освоить на практике применение режима однократного гаммирования.

---

#  
За-  
да-  
ние

---

Нужно

по-

до-

братъ

ключ,

что-

бы

по-

лу-

чить

со-

об-

ще-

ние

«С

Но-

вым

Го-

дом,

дру-

зья!»..

Раз-

ра-

бо-

та-

ем

при-

ло-

же-

ние,

поз-

во-

ляю-

щее

шиф-

ро-

вать

и де-

шиф-

ро-

вать

дан-

ные

в ре-

жи-

ме

о<sub>2</sub>

но-

крат-

ного

гам-

ми-

ро-

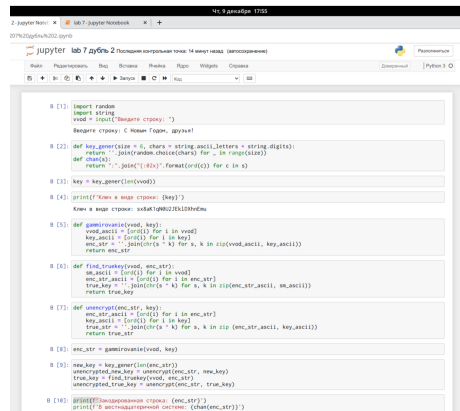
ва-

ния

---

## Выполнение лабораторной работы

1. Перейдем к написанию кода программы (рис.1).



```
0 (1) import random
import string
rand = random.choice(string)

Введите строку: С Новым Годом, друзья!

0 (2) def key_gen(size = 6, chars = string.ascii_letters + string.digits):
    return ''.join(random.choice(chars) for _ in range(size))
def chaxor(s):
    return ''.join(chr(ord(c) ^ ord(k)) for c, k in zip(s, key))

0 (3) key = key_gen(len(rand))

0 (4) print('Ключ к шифру: {}'.format(key))

Ключ к шифру: 8x8a1q00225188080w

0 (5) def gammatize(s, key):
    rand_key = rand
    key_ascii = [ord(c) for c in rand]
    key_ascii = [ord(c) for c in key]
    enc_str = ''
    for i, k in zip(rand_key, key_ascii):
        enc_str += chr(ord(s[i]) ^ ord(k))
    return enc_str

0 (6) def decode(enc_str, key):
    key_ascii = [ord(c) for c in key]
    enc_str = [ord(c) for c in enc_str]
    true_key = ''
    for i, k in zip(enc_str, key_ascii):
        true_key += chr(ord(enc_str[i]) ^ ord(k))
    return true_key

0 (7) def unencrypt(enc_str, key):
    key_ascii = [ord(c) for c in key]
    enc_str = [ord(c) for c in enc_str]
    true_key = ''
    for i, k in zip(enc_str, key_ascii):
        true_key += chr(ord(enc_str[i]) ^ ord(k))
    return true_key

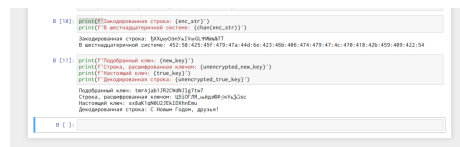
0 (8) enc_str = gammatize(rand, key)

0 (9) new_key = key_gen(len(enc_str))
unencrypted_new_key = unencrypt(enc_str, new_key)
true_key = rand
unencrypted_true_key = unencrypt(enc_str, true_key)

0 (10) print('Зашифрованная строка: {}'.format(enc_str))
print('В зашифрованной строке: {}'.format(enc_str))
```

Рис. 1: Рис.1. Начало программы.

2. Окончание программы, вывод (рис.2).



```
0 (10) print('Зашифрованная строка: {}'.format(enc_str))
print('В зашифрованной строке: {}'.format(enc_str))

Зашифрованная строка: 8x8a1q00225188080w
В зашифрованной строке: 8x8a1q00225188080w

0 (11) print('Подобный ключ: {}'.format(new_key))
print('Подобный ключ: {}'.format(new_key))
print('Подобная строка: {}'.format(unencrypted_true_key))

Подобный ключ: 8x8a1q00225188080w
Подобная строка: 8x8a1q00225188080w
В зашифрованной строке: 8x8a1q00225188080w
В зашифрованной строке: 8x8a1q00225188080w
```

Рис. 2: Рис.2. Конец программы, вывод.

3. Ответы на контрольные вопросы:

- 1) Поясните смысл однократного гаммирования. Гаммирование – это наложение (снятие) на открытые (зашифрованные) данные криптографической гаммы, то есть последовательности элементов данных, вырабатываемых с помощью некоторого криптографического алгоритма, для получения зашифрованных (открытых) данных. Однократное гаммирование – это когда каждый символ попарно с символом ключа складываются по модулю 2 (XOR).

4. 2) Перечислите недостатки однократного гаммирования. Недостатки: Размер ключевого материала должен совпадать с размером передаваемых сообщений. Также необходимо иметь эффективные процедуры для выработки случайных равновероятных двоичных последовательностей и специальную службу для развоза огромного количества ключей. А ещё, если одну и ту же гамму использовать дважды для разных сообщений, то шифр из совершенно стойкого превращается в «совершенно нестойкий» и допускает дешифрование практически вручную.
- 

5. 3) Перечислите преимущества однократного гаммирования. Достоинства: С точки зрения теории криптоанализа метод шифрования случайной однократной равновероятной гаммой той же длины, что и открытый текст, является невскрываемым. Кроме того, даже раскрыв часть сообщения, дешифровщик не сможет хоть сколько-нибудь поправить положение - информация о вскрытом участке гаммы не дает информации об остальных ее частях. К достоинствам также можно отнести простоту реализации и удобство применения.
- 

6. 4) Почему длина открытого текста должна совпадать с длиной ключа? Потому что каждый символ открытого текста должен складываться с символом ключа попарно.
- 

7. 5) Какая операция используется в режиме однократного гаммирования, назовите её особенности? В режиме однократного гаммирования используется сложение по модулю 2 (XOR) между элементами гаммы и элементами подлежащего сокрытию текста. Особенность заключается в том, что этот алгоритм шифрования является симметричным. Поскольку двойное прибавление одной и той же величины по модулю 2 восстанавливает исходное значение, шифрование и расшифрование выполняется одной и той же программой.
- 

8. 6) Как по открытому тексту и ключу получить шифротекст? Если известны ключ и открытый текст, то задача нахождения шифротекста заключается в применении к каждому символу открытого текста определенного правила. Размерности открытого текста и ключа должны совпадать, и полученный шифротекст будет такой же длины.
-

9. 7) Как по открытому тексту и шифротексту получить ключ? Если известны шифротекст и открытый текст, то задача нахождения ключа решается также в соответствии с правилом, а именно, обе части равенства необходимо сложить по модулю 2
- 
10. 8) В чем заключаются необходимые и достаточные условия абсолютной стойкости шифра? Необходимые и достаточные условия абсолютной стойкости шифра: Полная случайность ключа; Равенство длин ключа и открытого текста; Однократное использование ключа.

---

#  
Вы-  
во-  
ды

---

В  
ходе  
вы-  
пол-  
не-  
ния  
ла-  
бо-  
ра-  
тор-  
ной  
ра-  
бо-  
ты я  
изу-  
чи-  
ла  
тео-  
рию  
и  
осво-  
ила  
на  
прак-  
тике  
при-  
ме-  
не-  
ние  
ре-  
жи-  
ма  
од-  
но-  
крат-  
ного  
гам-  
ми-  
ро-  
ва-  
ния.

---

Спасибо за внимание!