

Отчёт по лабораторной работе №7

Элементы криптографии. Однократное гаммирование

Захарова Софья Михайловна

Оглавление

Цель работы	5
Задание	6
Выполнение лабораторной работы	7
Выводы	10

Список таблиц

Список иллюстраций

0.1	Рис.1. Начало программы.	7
0.2	Рис.2. Конец программы, вывод.	8

Цель работы

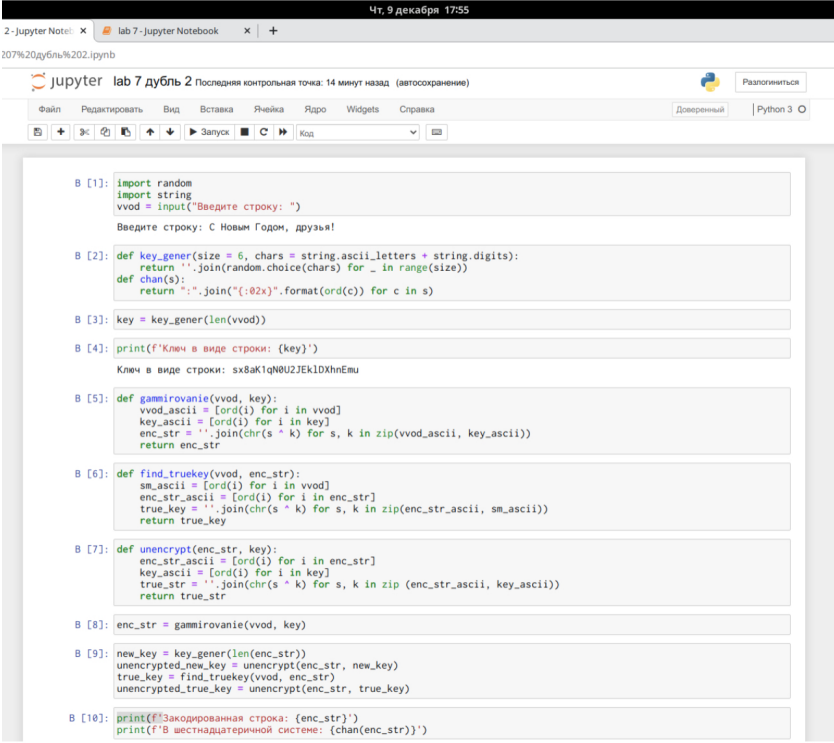
Освоить на практике применение режима однократного гаммирования.

Задание

Нужно подобрать ключ, чтобы получить сообщение «С Новым Годом, друзья!».
Разработаем приложение, позволяющее шифровать и дешифровать данные в режиме однократного гаммирования.

Выполнение лабораторной работы

Перейдем к написанию кода программы (рис.1).



```
B [1]: import random
import string
vvod = input("Введите строку: ")
Введите строку: С Новым Годом, друзья!

B [2]: def key_gener(size = 6, chars = string.ascii_letters + string.digits):
    return ''.join(random.choice(chars) for _ in range(size))
def chan(s):
    return "".join("{:02x}".format(ord(c)) for c in s)

B [3]: key = key_gener(len(vvod))

B [4]: print(f'Ключ в виде строки: {key}')
Ключ в виде строки: sx8aK1qN0U2JEk1DXhEму

B [5]: def gammirovane(vvod, key):
    vvod_ascii = [ord(i) for i in vvod]
    key_ascii = [ord(i) for i in key]
    enc_str = ''.join(chr(s ^ k) for s, k in zip(vvod_ascii, key_ascii))
    return enc_str

B [6]: def find_truekey(vvod, enc_str):
    sm_ascii = [ord(i) for i in vvod]
    enc_str_ascii = [ord(i) for i in enc_str]
    true_key = ''.join(chr(s ^ k) for s, k in zip(enc_str_ascii, sm_ascii))
    return true_key

B [7]: def unencrypt(enc_str, key):
    enc_str_ascii = [ord(i) for i in enc_str]
    key_ascii = [ord(i) for i in key]
    true_str = ''.join(chr(s ^ k) for s, k in zip(enc_str_ascii, key_ascii))
    return true_str

B [8]: enc_str = gammirovane(vvod, key)

B [9]: new_key = key_gener(len(enc_str))
unencrypted_new_key = unencrypt(enc_str, new_key)
true_key = find_truekey(vvod, enc_str)
unencrypted_true_key = unencrypt(enc_str, true_key)

B [10]: print(f'Закодированная строка: {enc_str}')
print(f'В шестнадцатеричной системе: {chan(enc_str)}')
```

Рис. 0.1: Рис.1. Начало программы.

Окончание программы, вывод (рис.2).

```
В [10]: print(f'Закодированная строка: {enc_str}')
print(f'В шестнадцатеричной системе: {hex(enc_str)}')
Закодированная строка: b'XxUyOenYzIVoGLYwbaTT'
В шестнадцатеричной системе: 452:58:425:45f:479:47a:44d:6e:423:46b:486:474:479:47:4c:470:418:42b:459:409:422:54

В [11]: print(f'Подобранный ключ: {new_key}')
print(f'Строка, расшифрованная ключом: {unencrypted_new_key}')
print(f'Настоящий ключ: {true_key}')
print(f'Декодированная строка: {unencrypted_true_key}')
Подобранный ключ: tmgKjab1JR2CwNj1g7tw7
Строка, расшифрованная ключом: Ц510ГЛЯ_мйдз0#скVz3Gsc
Настоящий ключ: x8BaK1qN0U2JEK1DXhneМи
Декодированная строка: С Новым Годом, друзья!

В [ ]:
```

Рис. 0.2: Рис.2. Конец программы, вывод.

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).

- 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.
- 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 невоскрываемым. 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.

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

Выводы

В ходе выполнения лабораторной работы я изучила теорию и освоила на практике применение режима однократного гаммирования.