

Отчет по лабораторной работе №8

**Элементы криптографии. Шифрование (кодирование) различных
исходных текстов одним ключом**

Дарья Эдуардовна Ибатулина

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
5	Выводы	10
6	Ответы на контрольные вопросы	11
	Список литературы	12

Список иллюстраций

4.1	Код программы, написанной на языке программирования Python для выполнения задания	9
-----	------------------------------------------------------------------------------------------------	---

Список таблиц

1 Цель работы

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

2 Задание

Два текста кодируются одним ключом (однократное гаммирование). Требуется не зная ключа и не стремясь его определить, прочесть оба текста. Необходимо разработать приложение, позволяющее шифровать и дешифровать тексты P_1 и P_2 в режиме однократного гаммирования. Приложение должно определить вид шифротекстов C_1 и C_2 обоих текстов P_1 и P_2 при известном ключе. Необходимо определить и выразить аналитически способ, при котором злоумышленник может прочесть оба текста, не зная ключа и не стремясь его определить.

3 Теоретическое введение

Исходные данные.

Две телеграммы Центра:

P1 = НаВашиходящийот1204

P2 = ВСеверныйфилиалБанка

Ключ Центра длиной 20 байт: K = 05 0C 17 7F 0E 4E 37 D2 94 10 09 2E 22 57 FF C8
0B B2 70 54

Шифротексты обеих телеграмм можно получить по формулам режима однократного гаммирования:

$$C1 = P1 \boxtimes K, C2 = P2 \boxtimes K.$$

Открытый текст можно найти, зная шифротекст двух телеграмм, зашифрованных одним ключом. Для этого оба равенства складываются по модулю 2. Тогда с учётом свойства операции XOR:

$$1 \boxtimes 1 = 0, 1 \boxtimes 0 = 1$$

получаем:

$$C1 \boxtimes C2 = P1 \boxtimes K \boxtimes P2 \boxtimes K = P1 \boxtimes P2.$$

Предположим, что одна из телеграмм является шаблоном — т.е. имеет текст фиксированный формат, в который вписываются значения полей. Допустим, что злоумышленнику этот формат известен. Тогда он получает достаточно много пар C1 \boxtimes C2 (известен вид обеих шифровок). Тогда зная P1 и учитывая выше приведенные формулы имеем:

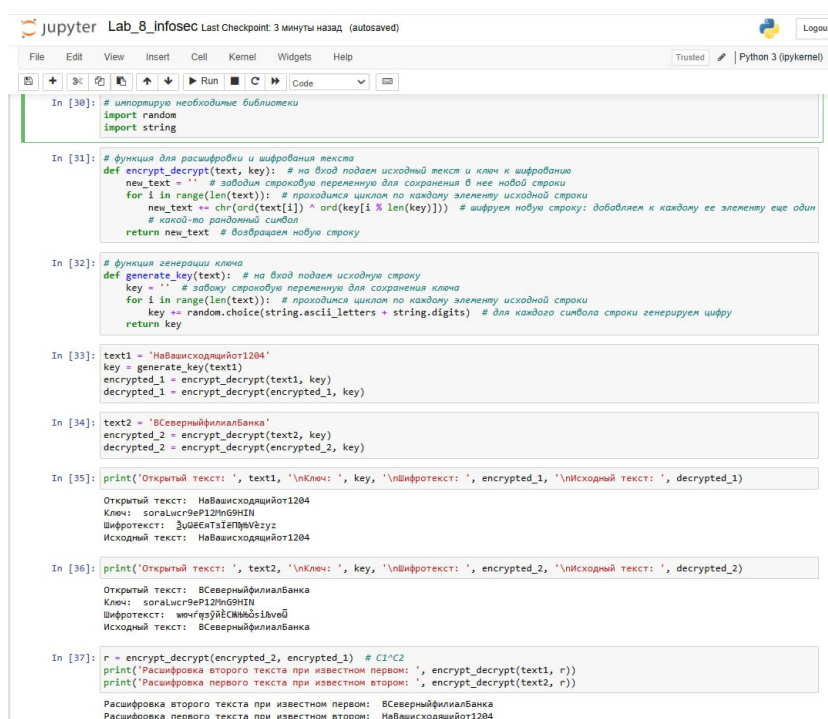
$$C1 \boxtimes C2 \boxtimes P1 = P1 \boxtimes P2 \boxtimes P1 = P2.$$

Таким образом, злоумышленник получает возможность определить те символы

сообщения P2, которые находятся на позициях известного шаблона сообщения P1. В соответствии с логикой сообщения P2, злоумышленник имеет реальный шанс узнать ещё некоторое количество символов сообщения P2. Затем вновь используется предыдущая формула с подстановкой вместо P1 полученных на предыдущем шаге новых символов сообщения P2. И так далее. Действуя подобным образом, злоумышленник даже если не прочитает оба сообщения, то значительно уменьшит пространство их поиска [0].

4 Выполнение лабораторной работы

Я написала на Питоне код программы, позволяющий шифровать различные исходные тексты одним ключом (рис. [4.1]):



```
Jupyter Lab_8_infosec Last Checkpoint: 3 минуты назад (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Python 3 (ipykernel)

In [30]: # импортируем необходимые библиотеки
import random
import string

In [31]: # функция для расшифровки и шифрования текста
def encrypt_decrypt(text, key): # на вход подает исходный текст и ключ к шифрованию
    new_text = '' # зададим строковую переменную для сохранения в нее новой строки
    for i in range(len(text)): # проходимся циклом по каждому элементу исходной строки
        new_text += chr(ord(text[i]) ^ ord(key[i % len(key)])) # шифруем новую строку: добавляем к каждому ее элементу еще один
        # какой-то случайный символ
    return new_text # возвращаем новую строку

In [32]: # функция генерации ключа
def generate_key(text): # на вход подает исходную строку
    key = '' # зададим строковую переменную для сохранения ключа
    for i in range(len(text)): # проходимся циклом по каждому элементу исходной строки
        key += random.choice(string.ascii_letters + string.digits) # для каждого символа строки генерируем цифру
    return key

In [33]: text1 = 'НаВашисходящий1204'
key = generate_key(text1)
encrypted_1 = encrypt_decrypt(text1, key)
decrypted_1 = encrypt_decrypt(encrypted_1, key)

In [34]: text2 = 'ВСеверныйФилиалБанка'
encrypted_2 = encrypt_decrypt(text2, key)
decrypted_2 = encrypt_decrypt(encrypted_2, key)

In [35]: print('Открытый текст: ', text1, '\nКлюч: ', key, '\nШифротекст: ', encrypted_1, '\nИсходный текст: ', decrypted_1)

Открытый текст:  НаВашисходящий1204
Ключ:  s0ga1nc9eP12HnG9HnH
Шифротекст:  3u06eKaTaePmV6zuZ
Исходный текст:  НаВашисходящий1204

In [36]: print('Открытый текст: ', text2, '\nКлюч: ', key, '\nШифротекст: ', encrypted_2, '\nИсходный текст: ', decrypted_2)

Открытый текст:  ВСеверныйФилиалБанка
Ключ:  s0ga1nc9eP12HnG9HnH
Шифротекст:  w0n4gz9y8ECM0b0s1kve0
Исходный текст:  ВСеверныйФилиалБанка

In [37]: r = encrypt_decrypt(encrypted_2, encrypted_1) # C1^C2
print('Расшифровка второго текста при известном первом: ', encrypt_decrypt(text1, r))
print('Расшифровка первого текста при известном втором: ', encrypt_decrypt(text2, r))

Расшифровка второго текста при известном первом:  ВСеверныйФилиалБанка
Расшифровка первого текста при известном втором:  НаВашисходящий1204
```

Рис. 4.1: Код программы, написанной на языке программирования Python для выполнения задания

5 Выводы

В результате выполнения данной лабораторной работы я научилась шифровать различные исходные тексты одним ключом.

6 Ответы на контрольные вопросы

1. **Как, зная один из текстов (P1 или P2), определить другой, не зная при этом ключа?** - Для определения другого текста (P2) можно просто взять зашифрованные тексты $C1 \boxtimes C2$, далее применить XOR к ним и к известному тексту: $C1 \boxtimes C2 \boxtimes P1 = P2$.
2. **Что будет при повторном использовании ключа при шифровании текста?** - При повторном использовании ключа мы получим дешифрованный текст.
3. **Как реализуется режим шифрования однократного гаммирования одним ключом двух открытых текстов?** - Режим шифрования однократного гаммирования одним ключом двух открытых текстов осуществляется путем XOR-ирования каждого бита первого текста с соответствующим битом ключа или второго текста.
4. **Перечислите недостатки шифрования одним ключом двух открытых текстов** - Недостатки шифрования одним ключом двух открытых текстов включают возможность раскрытия ключа или текстов при известном открытом тексте.
5. **Перечислите преимущества шифрования одним ключом двух открытых текстов** - Преимущества шифрования одним ключом двух открытых текстов включают использование одного ключа для зашифрования нескольких сообщений без необходимости создания нового ключа и выделения на него памяти.

Список литературы

[0] Методические материалы курса