Отчет по лабораторной работе №7

Дисциплина: Основы информационной безопасности

Иванов Сергей Владимирович

Содержание

# 1 Цель работы

Освоить на практике применение режима однократного гаммирования.

# 2 Задание

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

1. Определить вид шифротекста при известном ключе и известном открытом тексте.
2. Определить ключ, с помощью которого шифротекст может быть преобразован в некоторый фрагмент текста, представляющий собой один из возможных вариантов прочтения открытого текста.

# 3 Выполнение лабораторной работы

Я выбрал для написания программы язык Python. Для начала напишем функцию для генерации случайного ключа (рис. 1)

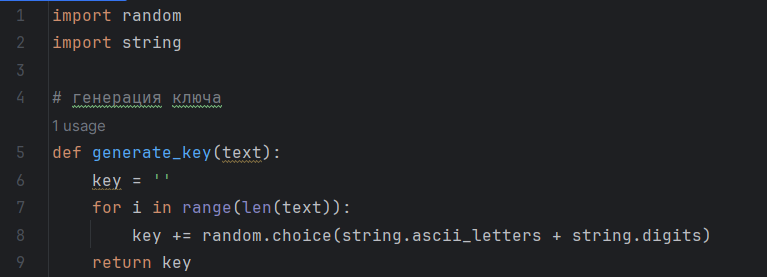


Рис. 1: Функция генерации ключа

Далее необходимо определить вид шифротекста при известном ключе и известном открытом тексте. Т.к операция XOR искючает сама себя, пишу одну функцию для шифровнаия и дешифрования текста(рис. 2).

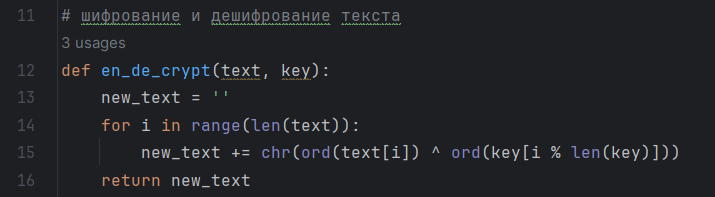


Рис. 2: Функция шифрования и дешифрования

Далее нужно определить ключ, с помощью которого шифротекст может быть преобразован в некоторый фрагмент текста, представляющий собой один из возможных вариантов прочтения открытого текста. Для этого пишу функцию для поиска возможных ключей для фрагмента текста. (рис. 3).

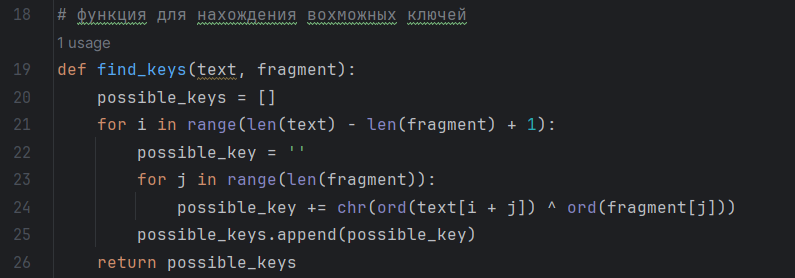


Рис. 3: Поиск возможных ключей

Проверяем всех функций. Убеждаемся, все работает корректно. (рис. 4).



Рис. 4: Проферка работы программы

Листинг программы

import random  
import string  
  
# генерация ключа  
def generate\_key(text):  
 key = ''  
 for i in range(len(text)):  
 key += random.choice(string.ascii\_letters + string.digits)  
 return key  
  
# шифрование и дешифрование текста  
def en\_de\_crypt(text, key):  
 new\_text = ''  
 for i in range(len(text)):  
 new\_text += chr(ord(text[i]) ^ ord(key[i % len(key)]))  
 return new\_text  
  
# функция для нахождения вохможных ключей  
def find\_keys(text, fragment):  
 possible\_keys = []  
 for i in range(len(text) - len(fragment) + 1):  
 possible\_key = ''  
 for j in range(len(fragment)):  
 possible\_key += chr(ord(text[i + j]) ^ ord(fragment[j]))  
 possible\_keys.append(possible\_key)  
 return possible\_keys  
  
text = 'С новым годом, друзья!'  
key = generate\_key(text)  
en\_crypt = en\_de\_crypt(text, key)  
de\_crypt = en\_de\_crypt(en\_crypt, key)  
find\_keys\_text = find\_keys(en\_crypt, 'С новым ')  
  
print("Открытый текст:", text, "\nКлюч:", key, "\nШифротекст:", en\_crypt, "\nИсходный текст:", de\_crypt)  
print("Возможные ключи:", find\_keys\_text)  
print("Расшифрованный фрагмент:", en\_de\_crypt(en\_crypt, find\_keys\_text[0]))

# 4 Ответы на контрольные вопросы

1. Поясните смысл однократного гаммирования. - Однократное гаммирование - это метод шифрования, при котором каждый символ открытого текста гаммируется с соответствующим символом ключа только один раз.
2. Перечислите недостатки однократного гаммирования. - Недостатки однократного гаммирования:

* Уязвимость к частотному анализу из-за сохранения частоты символов открытого текста в шифротексте.
* Необходимость использования одноразового ключа, который должен быть длиннее самого открытого текста.
* Нет возможности использовать один ключ для шифрования разных сообщений.

1. Перечислите преимущества однократного гаммирования. - Преимущества однократного гаммирования:

* Высокая стойкость при правильном использовании случайного ключа.
* Простота реализации алгоритма.
* Возможность использования случайного ключа.

1. Почему длина открытого текста должна совпадать с длиной ключа? - Длина открытого текста должна совпадать с длиной ключа, чтобы каждый символ открытого текста гаммировался с соответствующим символом ключа.
2. Какая операция используется в режиме однократного гаммирования, назовите её особенности? - В режиме однократного гаммирования используется операция XOR (исключающее ИЛИ), которая объединяет двоичные значения символов открытого текста и ключа для получения шифротекста. Особенность XOR - если один из битов равен 1, то результат будет 1, иначе 0.
3. Как по открытому тексту и ключу получить шифротекст? - Для получения шифротекста по открытому тексту и ключу каждый символ открытого текста гаммируется с соответствующим символом ключа с помощью операции XOR.
4. Как по открытому тексту и шифротексту получить ключ? - По открытому тексту и шифротексту невозможно восстановить действительный ключ, так как для этого нужна информация о каждом символе ключа.
5. В чем заключаются необходимые и достаточные условия абсолютной стойкости шифра - Необходимые и достаточные условия абсолютной стойкости шифра:

* Ключи должны быть случайными и использоваться только один раз.
* Длина ключа должна быть не менее длины самого открытого текста.
* Ключи должны быть храниться и передаваться безопасным способом.

# 5 Вывод

В ходе выполнения лабораторной работы мной было освоено на практике применение режима однократного гаммирования.