Лабораторная работа №7

Основы информационной безопасности

Кондрашова Анастасия Андреевна

Содержание

# 1 Теоретческое введение

Гаммирование представляет собой наложение (снятие) на открытые (зашифрованные) данные последовательности элементов других данных, полученной с помощью некоторого криптографического алгоритма, для получения зашифрованных (открытых) данных. Иными словами, наложение гаммы — это сложение её элементов с элементами открытого (закрытого) текста по некоторому фиксированному модулю, значение которого представляет собой известную часть алгоритма шифрования.

В соответствии с теорией криптоанализа, если в методе шифрования используется однократная вероятностная гамма (однократное гаммирование) той же длины, что и подлежащий сокрытию текст, то текст нельзя раскрыть. Даже при раскрытии части последовательности гаммы нельзя получить информацию о всём скрываемом тексте. Наложение гаммы по сути представляет собой выполнение операции сложения по модулю 2 (XOR) (обозначаемая знаком ⊕) между элементами гаммы и элементами подлежащего сокрытию текста.

# 2 Цель работы

Освоить основы шифрования через однократное гаммирование.

# 3 Выполнение лабораторной работы

Лабораторная работа выполнена на языке Python 3 в среде Jupiter Notebook.

1. Создаём функцию, которая осуществляет однократное гаммирование посредством побитового XOR

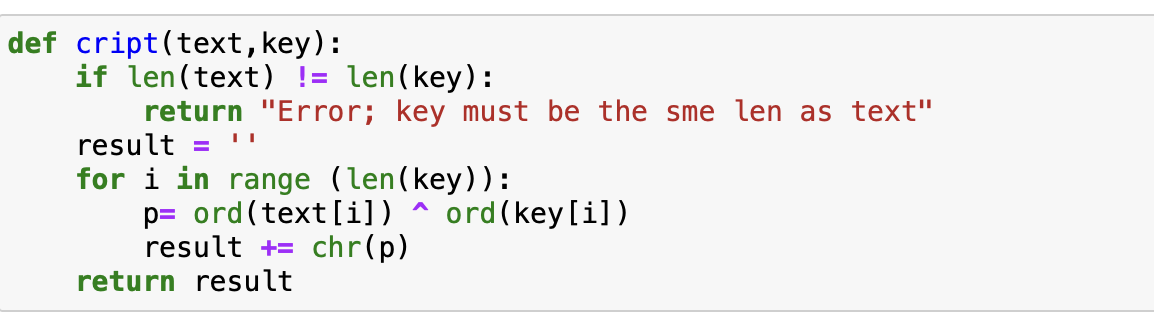


Рис. 1: Функция шифрования

1. Задаём текстовую строку и создаём случайный символьный ключ такой же длины



Рис. 2: Исходные данные

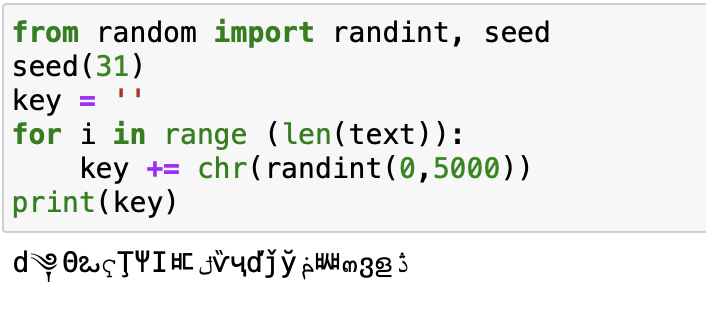
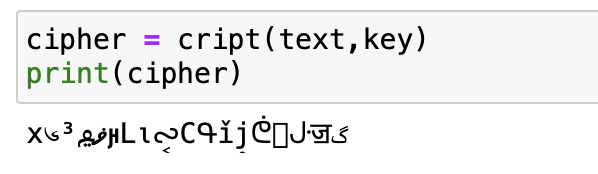


Рис. 3: Задание ключа

1. Запускаем функцию. В первом случае получаем зашифрованный текст. Далее, используя тот же самый ключ, осущвляем дешифровку текста. Так же, зная оригинальный текст и его шифорку, можем получить ключ. Все эти действия осуществляются через одну и ту же функцию. 

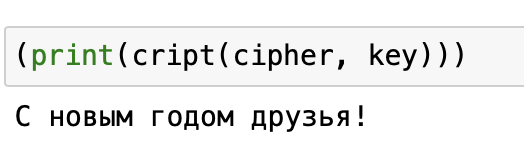


Рис. 4: Результат работы программы - дешифровка текста

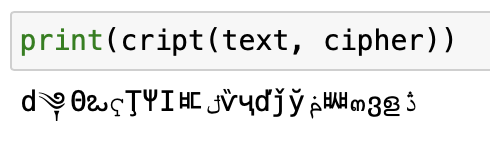


Рис. 5: Результат работы программы - ключ

# 4 Выводы

Я освоила основы шифрования через однократное гаммирование