Отчёт по лабораторной работе

Лабораторная №7

Дерябина Мария Сергеевна

Содержание

# Цель работы

Освоить на практике применение режима однократного гаммирования

# Теоретическая часть

Гаммирование представляет собой наложение (снятие) на открытые (зашифрованные) данные последовательности элементов других данных, полученной с помощью некоторого криптографического алгоритма, для получения зашифрованных (открытых) данных. Наложение гаммы — это сложение её элементов с элементами открытого (закрытого) текста по некоторому фиксированному модулю, значение которого представляет собой известную часть алгоритма шифрования. Наложение гаммы представляет собой выполнение операции сложения по модулю 2 (XOR) между элементами гаммы и элементами подлежащего сокрытию текста. Такой метод шифрования является симметричным, так как двойное прибавление одной и той же величины по модулю 2 восстанавливает исходное значение, а шифрование и расшифрование выполняется одной и той же программой. Если известны ключ и открытый текст, то задача нахождения шифротекста заключается в применении к каждому символу открытого текста следующего правила:

Ci = Pi ^ Ki,

где Ci — i-й символ получившегося зашифрованного послания, Pi — i-й символ открытого текста, Ki — i-й символ ключа, i = 1, m. Размерности открытого текста и ключа должны совпадать, и полученный шифротекст будет такой же длины. Если известны шифротекст и открытый текст, то задача нахождения ключа решается также, а именно, обе части равенства необходимо сложить по модулю 2 с Pi :

Ci ^ Pi = Pi ^ Ki ^ Pi = Ki,

Ki = Ci ^ Pi. Открытый текст имеет символьный вид, а ключ — шестнадцатеричное представление. Ключ также можно представить в символьном виде, воспользовавшись таблицей ASCII-кодов.

# Выполнение лабораторной работы

Написала программу на языке Python, позволяющую шифровать и дешифровать данные в режиме однократного гаммирования. Программа имеет 3 функции:

1. decode(cr\_message, key). Данная функция принимает зашифрованное сообщение и ключ (в виде строк с шестнадцатиричными значениями). Для каждого значения зашифрованного сообщения выполняется сложение по модулю 2 с сответствующим значением ключа. Функция возвращает строку с расшифрованным сообщением (рис. 1).

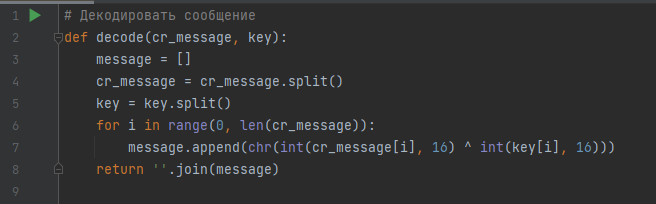


Figure 1: Функция для дешифрования сообщения

1. def encode(message, key). Данная функция принимает исходное сообщение и ключ. Каждый символ сообщения преобразовывется в число, соответствующее его коду в системе Unicode. Далее выполняется сложение по модулю 2 между получившимися кодами и соответствующими значениями ключа. Функция возвращает зашифрованное сообщение в виде строки с шестнадцатиричными значениями (рис. 2).

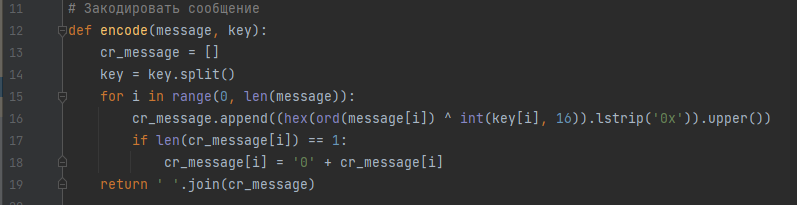


Figure 2: Функция для шифрования сообщения

1. get\_key(message, cr\_message). Данная функция принимает исходное сообщение и закодированное сообщение. Выполняется сложение по модулю 2 между кодами символов исходного сообщения и значениями закодированного сообщения. Функция возвращает ключ, с помощью которого исходный текст был закодирован (рис. 3).

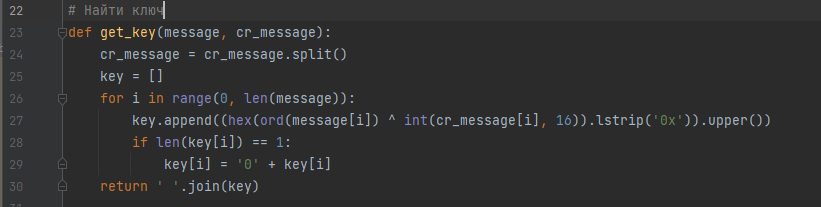


Figure 3: Функция для определения ключа шифрования

Написала код с вызовом функций для тестирования (рис. 4).

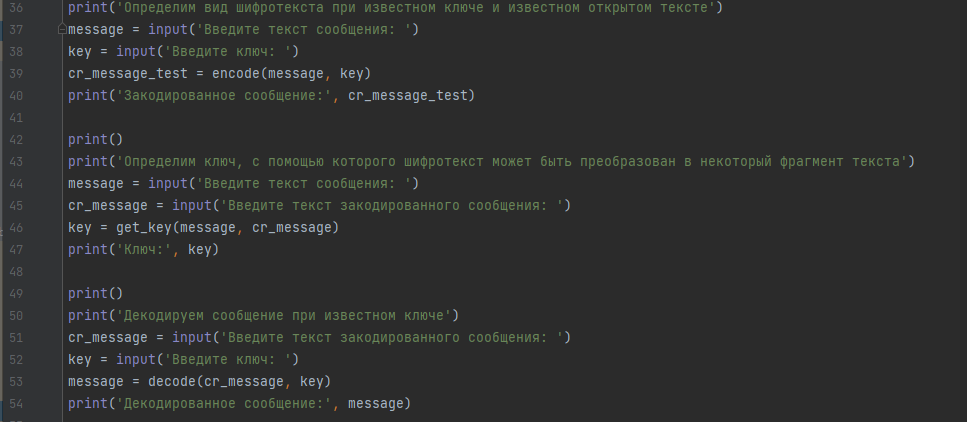


Figure 4: Вызов функций для тестирования

Протестировала программу на сообщении ‘С Новым Годом, друзья!’. При вызове различных функций убедилась, что программа корректно выполняет следующие задачи (рис. 5):

1. Определяет вид шифротекста при известном ключе и известном открытом тексте.
2. Определяет ключ, с помощью которого шифротекст может быть преобразован в некоторый фрагмент текста.
3. Выполняет дешифрование текста при известном ключе.

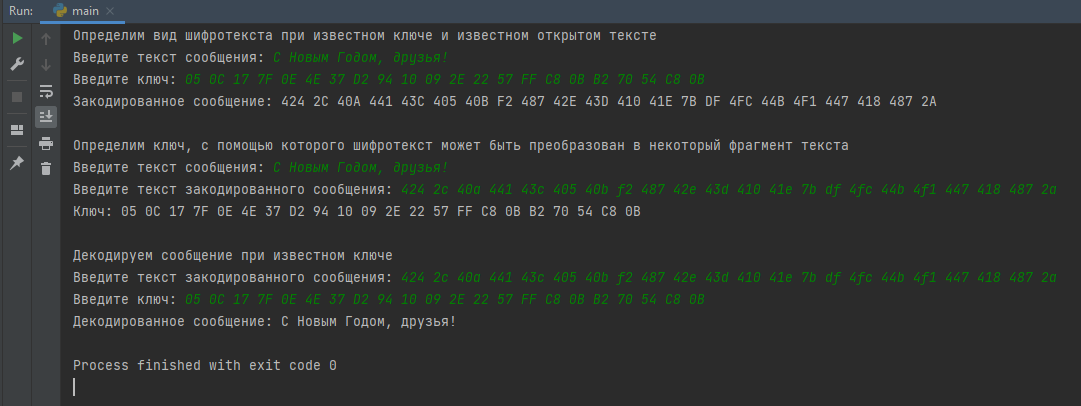


Figure 5: Тестирование программы

# Вывод

Я освоила на практике применение режима однократного гаммирования, разработала программу, выполняющую различные функции: шифрование и дешифрование текста, определение ключа.