Отчет по лабораторной работе №3

Шифрование гаммированием

Асеинова Елизавета Валерьевна

14 октября 2023

Содержание

# 1 Цель работы

Целью данной работы является ознакомление с шифрованием гаммированием, а также его программная реализация.

# 2 Задание

1. Изучить способ шифрования гаммированием.
2. Реализовать алгоритм шифрования гаммированием конечной гаммой на языке программирования Python.

# 3 Теоретическое введение

Из всех схем шифрования простейшей и наиболее надежной является схема однократного использования:

Формируется разрядная случайная двоичная последовательность - ключ шифра. Отправитель производит побитовое сложение по модулю два () ключа и разрядной двоичной последовательности , соответствующей посылаемому сообщению:

где - й бит исходного текста, - й бит ключа, - операция побитового сложения (XOR), - й бит получившейся криптограммы: .

Операция побитного сложения является обратимой, то есть , поэтому дешифрование осуществляется повторным применением операции к криптограмме:

Гаммирование - процедура наложения при помощи некоторой функции на исходный текст гаммы шифра, то есть псевдослучайной последовательности (ПСП) с выходом генератора . Псевдослучайная последовательность по своим статистическим свойствам неотличима от случайной последовательности, но является детерминированной, то есть известен алгоритм ее формирования. Обычно в качестве функции берется операция поразрядного сложения по модулю два или по модулю ( - число букв алфавита открытого текста) [1]

Простейший генератор псевдослуайной последовательности можно представить рекуррентным соотношением:

где - i-й член последовательности псевдослучайных чисел, - ключевые параметры. Такая последовательность состоит из целых чисел от 0 до m-1. Если элементы и совпадут, то совпадут и последующие участки: . Таким образом, ПСП является периодической. Знание периода гаммы существенно облегчает криптоанализ. Максимальная длина периода равна m. Для ее достижения необходимо удовлетворить следующим условиям:

* b и m - взаимно простые числа;
* a-1 делится на любой простой делитель числа m;
* a-1 кратно 4, если m кратно 4.

# 4 Ход выполнения лабораторной работы

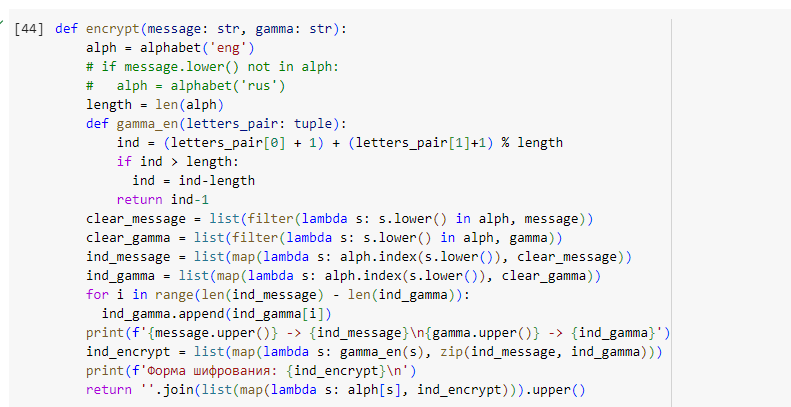
Для реализации шифров перестановки будем использовать среду JupyterLab. Выполним необходимую задачу.

1. Задаем функцию определения алфавита для последующего шифрования



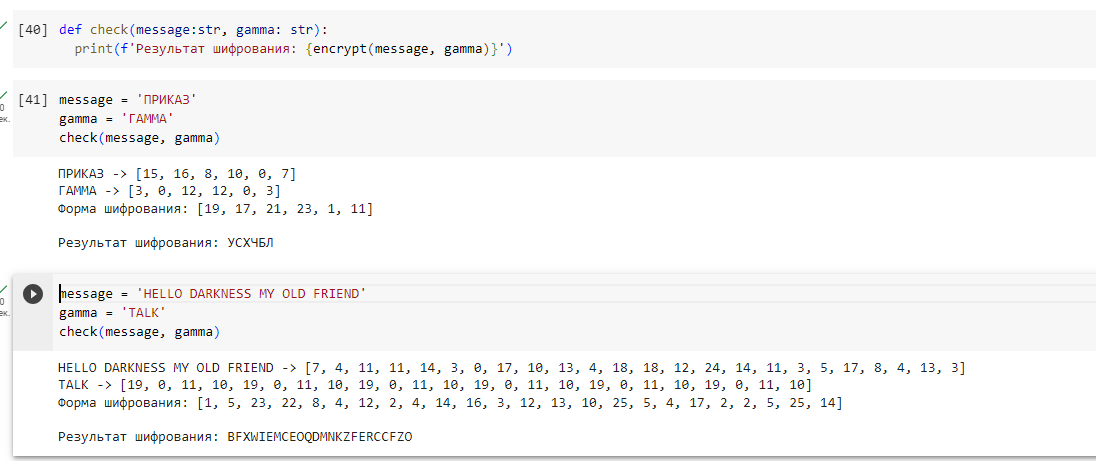
Алфавит для реализации шифров

1. Описываем функцию с принципом работы алгоритма шифрования конечной гаммой



Функция алгоритма шифрования конечной гаммой

1. Прописываем функцию для шифрования переданного текста. Задаем тестовые данные и вызываем функцию:



Реализация шифрования гаммированием на примере

Полученное сообщение аналогично приведенному в Методических материалах. Также на скриншоте можно увидеть пример на английском языке

# 5 Выводы

В рамках данной работы мы изучили и программно реализовали алгоритм шифрования гаммированием конечной гаммой.

# 6 Список литературы

1. Методические материалы курса[1]