Отчёт по лабораторной работе №7

Шифр гаммирования

Савченко Елизавета, НБИ-01-20

Содержание

List of Figures

1 Цель работы

Изучение алгоритма шифрования гаммированием

2 Теоретические сведения

2.1 Шифр гаммирования

Гаммирование – это наложение (снятие) на открытые (зашифрованные) данные криптографической гаммы, т.е. последовательности элементов данных, вырабатываемых с помощью некоторого криптографического алгоритма, для получения зашифрованных (открытых) данных.

Принцип шифрования гаммированием заключается в генерации гаммы шифра с помощью датчика псевдослучайных чисел и наложении полученной гаммы шифра на открытые данные обратимым образом (например, используя операцию сложения по модулю 2). Процесс дешифрования сводится к повторной генерации гаммы шифра при известном ключе и наложении такой же гаммы на зашифрованные данные. Полученный зашифрованный текст является достаточно трудным для раскрытия в том случае, если гамма шифра не содержит повторяющихся битовых последовательностей и изменяется случайным образом для каждого шифруемого слова. Если период гаммы превышает длину всего зашифрованного текста и неизвестна никакая часть исходного текста, то шифр можно раскрыть только прямым перебором (подбором ключа). В этом случае криптостойкость определяется размером ключа.

Метод гаммирования становится бессильным, если известен фрагмент исходного текста и соответствующая ему шифрограмма. В этом случае простым вычитанием по модулю 2 получается отрезок псевдослучайной последовательности и по нему восстанавливается вся эта последовательность.

Метод гаммирования с обратной связью заключается в том, что для получения сегмента гаммы используется контрольная сумма определенного участка шифруемых данных. Например, если рассматривать гамму шифра как объединение непересекающихся множеств H(j), то процесс шифрования можно пердставить следующими шагами:

1. Генерация сегмента гаммы H(1) и наложение его на соответствующий участок шифруемых данных.
2. Подсчет контрольной суммы участка, соответствующего сегменту гаммы H(1).
3. Генерация с учетом контрольной суммы уже зашифрованного участка данных следующего сегмента гамм H(2).
4. Подсчет контрольной суммы участка данных, соответствующего сегменту данных H(2) и т.д.

3 Выполнение работы

3.1 Реализация шифратора и дешифратора Python

```
import string
import random



def function1(text):
    return ' '.join(hex(ord(i))[2:] for i in text)

def function2(size):
    return ''.join(random.choice(string.ascii_letters+string.digits) for _ in ran

def function3(text,key):
    return ''.join(chr(a^b) for a,b in zip (text,key))

def function4(text,encrypt):
    return ''.join(chr(a^b) for a,b in zip (text,encrypt))



message = 'С Новым годом, друзья!'
key  = function2(len(message))
hex_key = function1(key)
```

```
print("Используем ключ: ", key)
print("Ключ в шестнадцатиричном виде: ", hex_key)
encrypt = function3([ord(i) for i in message], [ord(i) for i in key])
hex_encrypt = function1(encrypt)
print("Зашифрованное: ", hex_encrypt)
decrypt = function3([ord(i) for i in encrypt], [ord(i) for i in key])
print("Расшифрованное: ", decrypt)


compute_key = function4([ord(i) for i in message], [ord(i) for i in encrypt])
decrypt_compute_key = function3([ord(i) for i in encrypt], [ord(i) for i in key])
print("Исходный ключ: ", key)
print("Вариант прочтения открытого текста: ", decrypt_compute_key)
```

3.2 Контрольный пример

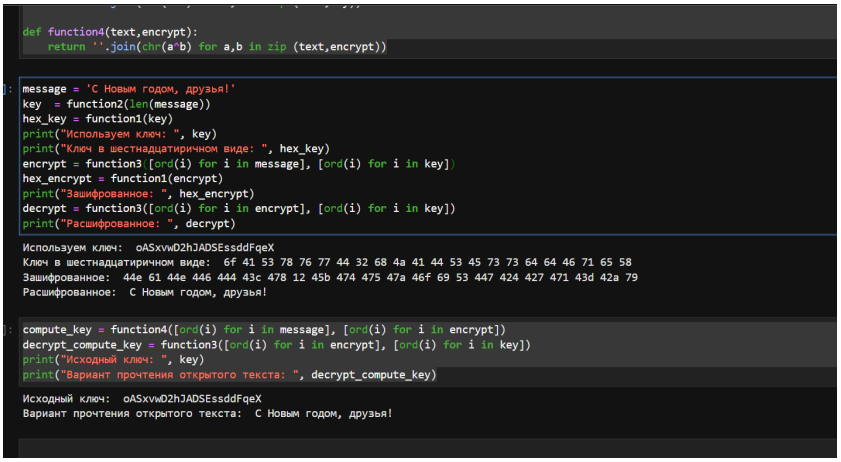

Figure 3.1: работа алгоритма гаммирования на практике

4 Выводы

Изучили алгоритмы шифрования на основе гаммирования

Список литературы

1. Шифрование методом гаммирования
2. Режим гаммирования в блочном алгоритме шифрования