Отчёт по лабораторной работе №7

дисциплина: Информационная безопасность

Быстров Глеб Андреевич

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

В данной лабораторной работе мне будет необходимо освоить на практике применение режима однократного гаммированияи.

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 Теоретическое введение

Гамми́рование, или Шифр XOR, — метод симметричного шифрования, заключающийся в «наложении» последовательности, состоящей из случайных чисел, на открытый текст. Последовательность случайных чисел называется гамма-последовательностью и используется для зашифровывания и расшифровывания данных. Суммирование обычно выполняется в каком-либо конечном поле. Например, в поле Галуа GF(2) суммирование принимает вид операции «исключающее ИЛИ (XOR)». [1].

Симметричное шифрование - это метод шифрования, при котором для защиты информации используется ключ, зная который любой может расшифровать или зашифровать данные.

Алгоритмы с симметричными ключами имеют очень высокую производительность. Криптография с симметричными ключами стойкая, что делает практически невозможным процесс дешифрования без знания ключа. При прочих равных условиях стойкость определяется длиной ключа. Так как для шифрования и дешифрования используется один и тот же ключ, при использовании таких алгоритмов требуются высоко надежные механизмы для распределения ключей. Ещё одна проблемой является безопасное распространение симметричных ключей. Алгоритмы симметричного шифрования используют ключи не очень большой длины и могут быстро шифровать большие объемы данных.

Гаммированием (gamma xoring) называется процесс «наложения» гамма-последовательности на открытые данные. Обычно это суммирование по какому-либо модулю, например, по модулю два, такое суммирование принимает

вид обычного «исключающего ИЛИ» суммирования.

Симметричное шифрование остаётся самым актуальным и криптографически гарантированными методом защиты информации. В симметричном шифровании, основанном на использовании составных ключей, идея состоит в том, что секретный ключ делится на две части, хранящиеся отдельно. Каждая часть сама по себе не позволяет выполнить дешифрование [2].

4 Выполнение лабораторной работы

1. Реализовал на языке Python программу для выполнения задания (рис. 4.1).

```
import random
from random import seed
import string

def func(text, key):
    if len(key) != len(text):
        return "Разная длина"
    ctext = ''
    for i in range(len(key)):
        ctext_s = ord(text[i]) ^ ord(key[i])
        ctext += chr(ctext_s)
    return ctext

text = "С Новым Годом, друзья!"

key = ''
seed(23)
for i in range (len(text)):
    key += random.choice(string.ascii_letters + string.digits)

ctext = func(text, key)

print('Зашифрованный текст:', ctext)
print('Открытый текст:', func(ctext, key))
print('Известный ключ:', func(text, ctext))
```

```
Зашифрованный текст: ЖхХэЇQњВцъЎчV[IwЭбVЭРо
Открытый текст: С Новым Годом, друзья!
Известный ключ: 7X8s51fbLtByHwiUmrCaoN
```

Рис. 4.1: Код программы

Этот код выполняет операцию шифрования и дешифрования текста с использованием операции исключающего ИЛИ (XOR) между каждым символом в исходном тексте и ключе. Вот, как работает код:

Сначала код импортирует необходимые модули:

- random: для генерации случайных символов.
- seed: для установки начального значения генератора случайных чисел, чтобы можно было воспроизводить результат.
- string: для доступа к наборам символов (букв и цифр).

Затем определена функция func, которая принимает два аргумента: text и key.

В функции func проверяется, равны ли длины text и key. Если они не равны, функция возвращает строку “Разная длина”.

Далее в цикле происходит шифрование каждой пары символов из text и key. Для этого:

- Символы из text и key преобразуются в их числовые коды с помощью функции ord.
- Затем выполняется операция XOR (^) между числовыми кодами символов, и результат записывается в переменную ctext_s.
- Полученный код символа ctext_s преобразуется обратно в символ с помощью функции chr.
- Зашифрованный символ добавляется к строке ctext.

Функция func возвращает строку ctext после завершения шифрования.

Затем определены исходный текст text и ключ key. Ключ генерируется следующим образом:

- Устанавливается начальное значение генератора случайных чисел с помощью seed(23), чтобы результаты были воспроизводимы.
- В цикле создается ключ, добавляя к нему случайные символы из множества букв и цифр.

Далее вызывается функция func с исходным текстом и ключом для зашифровки текста, и зашифрованный текст сохраняется в переменной ctext.

На экран выводится зашифрованный текст с помощью print.

Затем вызывается функция func дважды для дешифровки текста:

- Первый раз с зашифрованным текстом и ключом.
- Второй раз с исходным текстом и зашифрованным текстом.

Расшифрованный текст и исходный текст снова выводятся на экран с помощью print.

5 Выводы

В данной лабораторной работе мне успешно удалось освоить на практике применение режима однократного гаммированияи.

Список литературы

1. Гаммирование [Электронный ресурс]. 2023. URL: https://ru.wikipedia.org/wiki/%D0%93%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5.
2. Симметричное шифрование (гаммирование) [Электронный ресурс]. 2023. URL: http://engineering-science.ru/doc/187185.html.