Лабораторная работа 4

Тема:

“Шифрование произвольного файла методом гаммирования”

Выполнил:

Студент группы 22207 Гордеев Никита

Задание:

- При передаче в сетях больших объемов данных для шифрования используется метод гаммирования, связанный с наложением на исходное сообщение, рассматриваемое как последовательность битов, некоторой секретной гаммы (ключа шифрования) – псевдослучайной битовой последовательности той же длины.
- Над каждой парой битов с одинаковыми порядковыми номерами из исходной и псевдослучайной последовательностями выполняется операция суммирования по модулю два (XOR или исключающее ИЛИ – Е): Yi = Xi E Ki

Описание структуры файла-ключа (способ хранения в нем параметров ГПСЧ).

- Значения параметров a, b и c генерируются случайным образом, без использования стандартного генератора случайных чисел

Статистика

Тест 1 (Шифрование)

Значения ключей: a = 382579 b = 19563 c = 63268

Длина исходного тектса: 588 символа

Время выполнения программы составляет: 0.000997781753540039

Тест 2 (Дешифрование)

Значения ключей: a = 382579 b = 19563 c = 63268

Длина исходного тектса: 588 символа

Время выполнения программы составляет: 0.0009980201721191406

Тест 3 (Шифрование)

Значения ключей: a = 593665 b = 29713 c = 69864

Длина исходного тектса: 588 символа

Время выполнения программы составляет: 0.0010020732879638672

Тест 4 (Дешифрование)

Значения ключей: a = 593665 b = 29713 c = 69864

Длина исходного тектса: 588 символа

Время выполнения программы составляет: 0.0009951591491699219

Тест 5 (Шифрование)

Значения ключей: a = 848017 b = 41943 c = 77812

Длина исходного тектса: 4470 символа

Время выполнения программы составляет: 0.007971763610839844

Тест 6 (Дешифрование)

Значения ключей: a = 848017 b = 41943 c = 77812

Длина исходного тектса: 4470 символа

Время выполнения программы составляет: 0.0008951591491699219

Тест 7 (Шифрование)

Значения ключей: a = 689473 b = 34317 c = 72858

Длина исходного тектса: 24662 символа

Время выполнения программы составляет: 0.022934913635253906

Тест 8 (Дешифрование)

Значения ключей: a = 689473 b = 34317 c = 72858

Длина исходного тектса: 24662 символа

Время выполнения программы составляет: 0.03691244125366211

Тест 9 (Шифрование)

Значения ключей: a = 157615 b = 8747 c = 38602

Длина исходного тектса: 64774 символа

Время выполнения программы составляет: 0.020943880081176758

Тест 10 (Дешифрование)

Длина исходного тектса: 64774 символа

Время выполнения программы составляет: 0.019977331161499023

Фрагменты программы:

```
# ###################### НАСТРОЙКИ ОКРУЖЕНИЯ ######################
import struct
import time
import math
from datetime import datetime
import matplotlib.pyplot as plt
```

Рисунок 1, библиотеки

```
# ###################### ГЕНЕРАТОР ПСЕВДОСЛУЧАЙНЫХ ЧИСЕЛ ######################

# параметры для формулы ci = (a*ci-1+ b) (mod m)
def formula_parameters():
    n = 24
    m = 2 ** n
    return n, m

# Текущая микросекунда
def microsecond():
    return datetime.now().microsecond

# Текущий час
def hour():
    return datetime.now().hour

# Текущий день
def day():
    return datetime.now().day

# Текущий месяц
def month():
    return datetime.now().month

# Текущий год
def year():
    return datetime.now().year
```

Рисунок 2, параметры формулы

```
# Получение случайных a и b и c
def generate_abc():

    n, m = formula_parameters()

    # Параметр a. От времени суток, остаток.
    a = math.ceil(microsecond() / math.ceil(math.sqrt(day() / 57))) + \
        math.ceil(day() / math.ceil(year() * month())) + \
        math.ceil(34 / (7 + microsecond() % 24)) - year() * month()
    while a % 6 != 1:
        a += 1

    # Параметр b. От времени суток, остаток, НОД.
    b = year() * month() + 5 * \
        math.ceil(microsecond() / 104) + \
        math.ceil(31 / (12 + microsecond() % 14)) - year() * month()
    while b % 2 != 1 and math.gcd(b, m) != 1:
        b += 1

    # Параметр c. От времени суток.
    c = math.ceil(microsecond() / (hour() + 22)) + math.ceil(year() * (hour() + 3)) + \
        math.ceil(math.sqrt(day() / 43)) + \
        math.ceil(71 / (22 + microsecond() % 24)) + year() * month()

    # Вывод значений
    print("Значения ключей:", "\na =", a, "\nb =", b, "\nc =", c)

    return a, b, c
```

Рисунок 3, генератор a, b, c

```python
# ГПСЧ. Возвращает список или одно число. ci = (a*ci-1+ b) (mod m)
def linear_congruent_generator(a, b, c, sequence_length):

    n, m = formula_parameters()

    # degree_of_two = (microsecond() + day()) % 24
    degree_of_two = n

    # Последовательность длины 1 (ограничение сверху 4096)
    if sequence_length == 1:
        return [math.ceil((a * c + b) % 2**degree_of_two) % 255]

    # Массив символов исходного сообщения
    sequence_result = [0 for i in range(sequence_length + 1)]
    sequence_result[0] = math.ceil(c)

    for i in range(1, sequence_length + 1):
        sequence_result[i] = math.ceil((a * sequence_result[i-1] + b) % 2**degree_of_two) % 255

    # массив числовых представлений символов
    return sequence_result[1:sequence_length + 1]
```

Рисунок 4, генератор псевдо-случайных чисел

```python
# ##################### ШИФРОВАНИЕ #####################

# Гаммирования текста по ключу
def encrypt(text):
    # Место сохранения
    save_path = input("Введите название, под которым сохранить зашифрованное сообщение в формате {название}.txt: ")
    file_message = open(save_path, 'w', encoding="utf-8")
    message = ""

    # Получаем ключи
    key_path = input("Введите название файла-ключа (в формате {название}.key): ")
    file_keys = open(key_path, 'r', encoding="utf-8")
    keys = file_keys.read()
    file_keys.close()

    a, b, c = keys.split(" ")

    key = linear_congruent_generator(int(a), int(b), int(c), len(text))

    # Запуск секундомера
    time_start = time.time()

    # Гаммирование
    for i in range(len(text)):
        # ord = число символа Unicode. ^ = XOR
        simvol = ord(text[i]) ^ int(key[i])
        # chr = число в символ Unicode
        message += chr(simvol)

    # Остановка секундомера
    time_stop = time.time()

    print("Время выполнения программы составляет: ", time_stop - time_start)
    file_message.write(message)
    file_message.close()
```

Рисунок 5, шифрование

```python
# ##################### ИНТЕРФЕЙС ПОЛЬЗОВАТЕЛЯ #####################

def main():

    text = None

    while True:
        user_select = int(input(
            "Выберите режим:\n" +
            "(1): Сгенерировать и записать в файл ключ\n" +
            "(2): Зашифровать/дешифровать файл\n" +
            "(3): Выход\n" +
            "Ваш выбор: "))

        if user_select == 1:
            # Сгенирировать параметры a, b, c
            a, b, c = generate_abc()

            # записать ключ в файл
            key_path = input("Введите название, под которым сохранить файл-ключей (в формате {название}.key): ")
            file = open(key_path, 'w', encoding="utf-8")
            file.write(str(a) + ' ' + str(b) + " " + str(c))
            file.close()

        elif user_select == 2:
            try:
                # Получить файл
                save_path = input("Введите путь до файла с текстом (в формате {название}.txt): ")
                file = open(save_path, "r", encoding="utf-8")
                text = file.read()
                file.close()

                # Зашифровать / расшифровать файл
                encrypt(text)
            except:
                print("В начале необходимо сгенерировать и записать в файл ключ!")

        elif user_select == 3:
            # Выход
            exit(0)
```

Рисунок 6, интерфейс пользователя

Материалы:

Random int without importing 'random' // stackoverflow URL: https://stackoverflow.com/questions/22950768/random-int-without-importing-random (дата обращения: 07.12.2022).

Python Random Function without using random module // stackoverflow URL: https://stackoverflow.com/questions/28705965/python-random-function-without-using-random-module (дата обращения: 07.12.2022).

Clock drift // Wikipedia URL: https://en.wikipedia.org/wiki/Clock_drift#Random_number_generators (дата обращения: 07.12.2022).

Datetime current year and month in Python // stackoverflow URL: https://stackoverflow.com/questions/28189442/datetime-current-year-and-month-in-python (дата обращения: 07.12.2022).

Matplotlib.pyplot.hist() in Python // GeeksforGeeks URL: https://www.geeksforgeeks.org/matplotlib-pyplot-hist-in-python/ (дата обращения: 07.12.2022).

Python math.fmod() Method // w3schools URL: https://www.w3schools.com/python/ref_math_fmod.asp (дата обращения: 07.12.2022).

Math.ceil() // mdn web docs _ URL: https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Global_Objects/Math/ceil (дата обращения: 07.12.2022).

Writing printed output to a file (python) // stackoverflow URL: https://stackoverflow.com/questions/16190973/writing-printed-output-to-a-file-python (дата обращения: 07.12.2022).