

# **Отчёт по лабораторной работе 7**

**Элементы криптографии. Однократное гаммирование**

Алли Мохамед Заян НКНбд-01-18

# Содержание

## Contents

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Цель работы.....</b>                     | <b>5</b>  |
| <b>2</b> | <b>Теоретические сведения.....</b>          | <b>6</b>  |
| <b>3</b> | <b>Выполнение лабораторной работы .....</b> | <b>8</b>  |
| <b>4</b> | <b>Контрольные вопросы .....</b>            | <b>10</b> |
| <b>5</b> | <b>Выводы.....</b>                          | <b>12</b> |
| <b>6</b> | <b>Список литературы.....</b>               | <b>13</b> |

# List of Tables

## List of Figures

|     |   |   |
|-----|---|---|
| 3.1 | Блок функции для расчетов . . . . .                     | 8 |
| 3.2 | Задание 1. Получение шифротекста . . . . .              | 8 |
| 3.3 | Один из вариантов прочтения открытого текста: . . . . . | 9 |

# **1 Цель работы**

Освоить на практике применение режима однократного гаммирования.

## 2 Теоретические сведения

Предложенная Г. С. Вернамом так называемая «схема однократного использования (гаммирования)» является простой, но надёжной схемой шифрования данных.

*Гаммирование* представляет собой наложение (снятие) на открытые (зашифрованные) данные последовательности элементов других данных, полученной с помощью некоторого криптографического алгоритма, для получения зашифрованных (открытых) данных. Иными словами, наложение гаммы — это сложение её элементов с элементами открытого (закрытого) текста по некоторому фиксированному модулю, значение которого представляет собой известную часть алгоритма шифрования. В соответствии с теорией криптоанализа, если в методе шифрования используется однократная вероятностная гамма (однократное гаммирование) той же длины, что и подлежащий сокрытию текст, то текст нельзя раскрыть. Даже при раскрытии части последовательности гаммы нельзя получить информацию о всём скрываемом тексте. Наложение гаммы по сути представляет собой выполнение операции сложения по модулю 2 (XOR) (обозначаемая знаком  $\oplus$ ) между элементами гаммы и элементами подлежащего сокрытию текста. Напомним, как работает операция XOR над битами:  $0 \oplus 0 = 0$ ,  $0 \oplus 1 = 1$ ,  $1 \oplus 0 = 1$ ,  $1 \oplus 1 = 0$ . 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:

$$C_i = P_i \oplus K_i$$

где  $C_i$  —  $i$ -й символ получившегося зашифрованного послания,  $P_i$  —  $i$ -й символ открытого текста,  $K_i$  —  $i$ -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  $P_i$ :

$$C_i \oplus P_i = P_i \oplus K_i \oplus P_i = K_i, K_i = C_i \oplus P_i.$$

Открытый текст имеет символьный вид, а ключ — шестнадцатеричное представление. Ключ также можно представить в символьном виде, воспользовавшись таблицей ASCII-кодов. К. Шеннон доказал абсолютную стойкость шифра в случае, когда однократно используемый ключ, длиной, равной длине исходного сообщения, является фрагментом истинно случайной двоичной последовательности с равномерным законом распределения. Криптоалгоритм не даёт никакой информации об открытом тексте: при известном зашифрованном сообщении  $C$  все различные ключевые последовательности  $K$  возможны и равновероятны, а значит, возможны и любые сообщения  $P$ . Необходимые и достаточные условия абсолютной стойкости шифра: — полная случайность ключа; — равенство длин ключа и открытого текста; — однократное использование ключа. [1]

## 3 Выполнение лабораторной работы

### 1. Написал блок функции для расчетов. (рис. 3.1)

```
In [8]: import string
import random

In [9]: #перевод в шестнадцатичную систему.
def hexx(text):
    return ''.join(hex(ord(i))[2:] for i in text)
#генерирует случайный ключ.
def gen_key(size):
    return ''.join(random.choice(string.ascii_letters + string.digits) for _ in range(size))
#шифрует и дешифрует текст.
def encrypted(text, key):
    return ''.join(chr(a^b) for a, b in zip(text, key))
#создает ключ на основе текста и шифротекста.
def compute_key(text,encrypt):
    return ''.join(chr(a^b) for a,b in zip(text,encrypt))
```

Figure 3.1: Блок функции для расчетов

### 2. Определил вид шифротекста при известном ключе и известном открытом тексте. (рис. 3.2)

```
In [10]: message= 'С Новым Годом, друзья!'

key=gen_key(len(message))
hex_key=hexx(key)

print("Используемый ключ: ", key)

print("Ключ в шестнадцатичном виде: ",hex_key )

encrypt = encrypted([ord(i) for i in message], [ord(i) for i in key])
hex_encrypt=hexx(encrypt)

print("Зашифрованное сообщение: ",hex_encrypt )

decryptt = encrypted([ord(i) for i in encrypt], [ord(i) for i in key])

print("Расшифрованное сообщение: ",decryptt )

Используемый ключ:  ALToUvnEtSTJ6tGq19NzXT
Ключ в шестнадцатичном виде:  41 4c 54 6f 55 76 6e 45 74 53 54 4a 36 74 47 71 6c 39 4e 7a 78 54
Зашифрованное сообщение:  460 6c 449 451 467 43d 452 65 467 46d 460 474 40a 58 67 445 42c 47a 479 436 437 75
Расшифрованное сообщение:  С Новым Годом, друзья!
```

Figure 3.2: Задание 1. Получение шифротекста



3. Определил ключ, с помощью которого шифротекст может быть преобразован в некоторый фрагмент текста, представляющий собой один из возможных вариантов прочтения открытого текста. (рис. 3.3)

```
In [27]: compute_key = compute_key([ord(i) for i in message], [ord(i) for i in encrypt])
decrypt_compute_key= encrypted([ord(i) for i in encrypt], [ord(i) for i in key])
print("Исходный ключ",key)
print("Вариант прочтения лткрытого текста: ", decrypt_compute_key)

Исходный ключ FyBNONLU6hVcP92va4XlUC
Вариант прочтения лткрытого текста:  С Новым Годом, друзья!
```

Figure 3.3: Один из вариантов прочтения открытого текста:

## 4 Контрольные вопросы

1. Поясните смысл однократного гаммирования.

Гаммирование — метод симметричного шифрования, заключающийся в «наложении» последовательности, состоящей из случайных чисел, на открытый текст. Последовательность случайных чисел называется гаммапоследовательностью и используется для зашифровывания и расшифровывания данных.

2. Перечислите недостатки однократного гаммирования.

Ключ одного размера с сообщением, на один ключ используется только один текст.

3. Перечислите преимущества однократного гаммирования.

Простота и криптостойкость.

4. Почему длина открытого текста должна совпадать с длиной ключа?

Каждый символ текста попарно складывается с символом ключа.

5. Какая операция используется в режиме однократного гаммирования, назовите её особенности?

Сложение по модулю 2. Особенность в симметричности – операция при повторном применении дает исходный результат.

6. Как по открытому тексту и ключу получить шифротекст?

Сложить по модулю 2 каждый символ открытого текста и ключа.

7. Как по открытому тексту и шифротексту получить ключ?

Сложить по модулю 2 каждый символ открытого текста и шифротекста.

8. В чем заключаются необходимые и достаточные условия абсолютной стойкости шифра?

- полная случайность ключа;
- равенство длин ключа и открытого текста;
- однократное использование ключа.

## **5 Выводы**

Освоил на практике применение режима однократного гаммирования.

## 6 Список литературы

1. Д. С. Кулябов, А. В. Королькова, М. Н. Геворкян. Информационная безопасность компьютерных сетей: лабораторные работы. // Факультет физико-математических и естественных наук. М.: РУДН, 2015. 64 с.