

# Отчет по лабораторной работе №8

## Основы информационной безопасности

Авдадаев Джамал. Группа: НКАбд-01-23

### Содержание

1	Цель работы .....	1
2	Задание .....	1
3	Теоретическое введение.....	1
4	Выполнение лабораторной работы.....	2
5	Ответы на контрольные вопросы.....	3
6	Выводы.....	5
	Список литературы .....	5

### 1 Цель работы

Освоить на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом

### 2 Задание

Два текста кодируются одним ключом (однократное гаммирование). Требуется не зная ключа и не стремясь его определить, прочитать оба текста. Необходимо разработать приложение, позволяющее шифровать и дешифровать тексты  $P_1$  и  $P_2$  в режиме однократного гаммирования. Приложение должно определить вид шифротекстов  $C_1$  и  $C_2$  обоих текстов  $P_1$  и  $P_2$  при известном ключе; Необходимо определить и выразить аналитически способ, при котором злоумышленник может прочитать оба текста, не зная ключа и не стремясь его определить.

### 3 Теоретическое введение

Исходные данные.

Две телеграммы Центра:

$P_1$  = НаВашисходящийот1204

$P_2$  = ВСеверныйфилиалБанка

Ключ Центра длиной 20 байт: K = 05 0C 17 7F 0E 4E 37 D2 94 10 09 2E 22 57 FF C8 0B B2 70 54

Шифротексты обеих телеграмм можно получить по формулам режима однократного гаммирования:

$$C_1 = P_1 \oplus K,$$

$$C_2 = P_2 \oplus K. \quad (8.1)$$

Открытый текст можно найти, зная шифротекст двух телеграмм, зашифрованных одним ключом. Для это оба равенства (8.1) складываются по модулю 2. Тогда с учётом свойства операции XOR

$$1 \oplus 1 = 0, 1 \oplus 0 = 1 \quad (8.2)$$

получаем:

$$C_1 \oplus C_2 = P_1 \oplus K \oplus P_2 \oplus K = P_1 \oplus P_2.$$

Предположим, что одна из телеграмм является шаблоном — т.е. имеет текст фиксированный формат, в который вписываются значения полей. Допустим, что злоумышленнику этот формат известен. Тогда он получает достаточно много пар  $C_1 \oplus C_2$  (известен вид обеих шифровок). Тогда зная  $P_1$  и учитывая (8.2), имеем:

$$C_1 \oplus C_2 \oplus P_1 = P_1 \oplus P_2 \oplus P_1 = P_2. \quad (8.3)$$

Таким образом, злоумышленник получает возможность определить те символы сообщения  $P_2$ , которые находятся на позициях известного шаблона сообщения  $P_1$ . В соответствии с логикой сообщения  $P_2$ , злоумышленник имеет реальный шанс узнать ещё некоторое количество символов сообщения  $P_2$ . Затем вновь используется (8.3) с подстановкой вместо  $P_1$  полученных на предыдущем шаге новых символов сообщения  $P_2$ . И так далее. Действуя подобным образом, злоумышленник даже если не прочитает оба сообщения, то значительно уменьшит пространство их поиска. [1]

## 4 Выполнение лабораторной работы

Я выполнил свою лабораторную работу на языке программирования Python, используя функции, реализованные в лабораторной работе №7.

Используя функцию для генерации ключа, генерирую ключ, затем шифрую два разных текста одним и тем же ключом (рис. 1).

```

1 import random
2 import string
3
4 def generate_key_hex(text):
5     key = ''
6     for i in range(len(text)):
7         key += random.choice(string.ascii_letters + string.digits) #генерация цифры для
            каждого символа в тексте
8     return key
9
10 #для шифрования и дешифрования
11 def en_de_crypt(text, key):
12     new_text = ''
13     for i in range(len(text)): #проход по каждому символу в тексте
14         new_text += chr(ord(text[i]) ^ ord(key[i % len(key)]))
15     return new_text
16
17 t1 = 'на 51!'
18 key = generate_key_hex(t1)
19 en_t1 = en_de_crypt(t1, key)
20 de_t1 = en_de_crypt(en_t1, key)
21
22 t2 = "Закреть инфобез!!"
23 en_t2 = en_de_crypt(t2, key)
24 de_t2 = en_de_crypt(en_t2, key)
25
26 print('Открытый текст: ', t1, "\nКлюч: ", key, '\nШифротекст: ', en_t1, '\nИсходный
    текст: ', de_t1,)
27 print('Открытый текст: ', t2, "\nКлюч: ", key, '\nШифротекст: ', en_t2, '\nИсходный
    текст: ', de_t2,)
28
29 r = en_de_crypt(en_t2, en_t1) #C1^C2
30 print('Расшифровать второй текст, зная первый: ', en_de_crypt(t1, r))
31 print('Расшифровать первый текст, зная второй: ', en_de_crypt(t2, r))
32

```

**Рис. 1: Шифрование двух текстов**

Расшифровываю оба текста сначала с помощью одного ключа, затем предполагаю, что мне неизвестен ключ, но известен один из текстов и уже расшифровываю второй, зная шифротексты и первый текст (рис. 2).

```

print('Открытый текст: ', t1, "\nКлюч: ", key, '\nШифротекст: ', en_t1, '\nИсходный текст: ', de_t1,)
print('Открытый текст: ', t2, "\nКлюч: ", key, '\nШифротекст: ', en_t2, '\nИсходный текст: ', de_t2,)

r = en_de_crypt(en_t2, en_t1) #C1^C2
print('Расшифровать второй текст, зная первый: ', en_de_crypt(t1, r))
print('Расшифровать первый текст, зная второй: ', en_de_crypt(t2, r))

Открытый текст: на 51!
Ключ: uVQymj
Шифротекст: ш!aQ@fk
Исходный текст: на 51!
Открытый текст: Закреть инфобез!!
Ключ: uVQymj
Шифротекст: Б!ЖеМШйвшГефЪ!tv
Исходный текст: Закреть инфобез!!
Расшифровать второй текст, зная первый: Закрыт
Расшифровать первый текст, зная второй: на 51!на 51

```

**Рис. 2: Результат работы программы**

## Листинг программы 1

```

import random
import string

```

```

def generate_key_hex(text):
    key = ''
    for i in range(len(text)):
        key += random.choice(string.ascii_letters + string.digits) #генерация
цифры для каждого символа в тексте
    return key

#для шифрования и дешифрования
def en_de_crypt(text, key):
    new_text = ''
    for i in range(len(text)): #проход по каждому символу в тексте
        new_text += chr(ord(text[i]) ^ ord(key[i % len(key)]))
    return new_text

t1 = 'на 51!'
key = generate_key_hex(t1)
en_t1 = en_de_crypt(t1, key)
de_t1 = en_de_crypt(en_t1, key)

t2 = "Закрыть инфобез!!"
en_t2 = en_de_crypt(t2, key)
de_t2 = en_de_crypt(en_t2, key)

print('Открытый текст: ', t1, "\nКлюч: ", key, '\nШифротекст: ', en_t1,
'\nИсходный текст: ', de_t1,)

print('Открытый текст: ', t2, "\nКлюч: ", key, '\nШифротекст: ', en_t2,
'\nИсходный текст: ', de_t2,)

r = en_de_crypt(en_t2, en_t1) #C1^C2
print('Расшифровать второй текст, зная первый: ', en_de_crypt(t1, r))
print('Расшифровать первый текст, зная второй: ', en_de_crypt(t2, r))

```

## 5 Ответы на контрольные вопросы

1. Как, зная один из текстов ( $P_1$  или  $P_2$ ), определить другой, не зная при этом ключа? - Для определения другого текста ( $P_2$ ) можно просто взять зашифрованные тексты  $C_1 \oplus C_2$ , далее применить XOR к ним и к известному тексту:  $C_1 \oplus C_2 \oplus P_1 = P_2$ .
2. Что будет при повторном использовании ключа при шифровании текста? - При повторном использовании ключа мы получим дешифрованный текст.
3. Как реализуется режим шифрования однократного гаммирования одним ключом двух открытых текстов? - Режим шифрования однократного гаммирования одним ключом двух открытых текстов осуществляется путем XOR-ирования каждого бита первого текста с соответствующим битом ключа или второго текста.
4. Перечислите недостатки шифрования одним ключом двух открытых текстов - Недостатки шифрования одним ключом двух открытых текстов включают возможность раскрытия ключа или текстов при известном открытом тексте.
5. Перечислите преимущества шифрования одним ключом двух открытых текстов - Преимущества шифрования одним ключом двух открытых текстов включают использование одного ключа для зашифрования нескольких сообщений без необходимости создания нового ключа и выделения на него памяти.

## 6 Выводы

В ходе лабораторной работы я освоил навыки применения режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом.

## Список литературы

1. Кулябов Д. С. Г.М.Н. Королькова А. В. Лабораторная работа № 8. Элементы криптографии. Шифрование (кодирование) различных исходных текстов одним ключом [Электронный ресурс]. 2023. URL: [https://esystem.rudn.ru/pluginfile.php/2293724/mod\\_resource/content/2/008-lab\\_crypto-key.pdf](https://esystem.rudn.ru/pluginfile.php/2293724/mod_resource/content/2/008-lab_crypto-key.pdf).