

Отчёт по лабораторной работе №8

**Элементы криптографии. Шифрование (кодирование) различных
исходных текстов одним ключом**

Аминов Зулфикор Мирзокаримович

Содержание

1. Цель работы	3
2. Указание к работе	4
3. Выполнение работы	6
3.1. Программа на python	6
3.2. Результат работы	7
4. Выводы	8

1. Цель работы

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

2. Указание к работе

Исходные данные.

Две телеграммы Центра:

$P_1 = \text{НаВашиходящийот1204}$

$P_2 = \text{ВСеверныйфилиалБанка}$

Ключ Центра длиной 20 байт:

$K = 05\ 0C\ 17\ 7F\ 0E\ 4E\ 37\ D2\ 94\ 10\ 09\ 2E\ 22\ 57\ FF\ C8\ 0B\ B2\ 70\ 54$

Режим шифрования однократного гаммирования одним ключом двух видов открытого текста реализуется в соответствии со схемой, приведённой на рис. 8.1.

Шифротексты обеих телеграмм можно получить по формулам режима однократного гаммирования:

$$\begin{aligned} C_1 &= P_1 \oplus K, \\ C_2 &= P_2 \oplus K. \end{aligned} \tag{8.1}$$

Открытый текст можно найти в соответствии с (8.1), зная шифротекст двух телеграмм, зашифрованных одним ключом. Для это оба равенства (8.1)

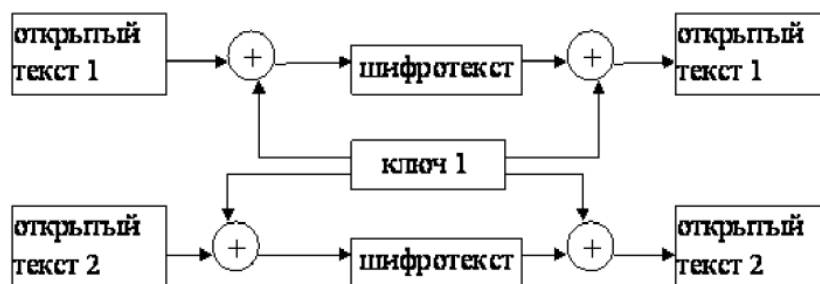


Рис. 2.1.: Общая схема шифрования двух различных текстов одним ключом

складываются по модулю 2. Тогда с учётом свойства операции XOR

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

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 $C_1 \boxtimes C_2$ (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.

3. Выполнение работы

3.1. Программа на python

```
import random

P_1 = "НаВашисходящийот1204"
P_2 = "ВСеверныйфилиалБанка"

def key(text):
    K = []
    for i in range(len(text)):
        K.append(random.randint(0, 2000))
    return K

def hex_key(key):
    h_k = []
    for i in key:
        h_k.append(hex(i))
    return h_k

def encod_and_decod(text, key):
    c = ""
    for i in range(len(text)):
```

```

        c += chr(ord(text[i]) ^ key[i])
    return c

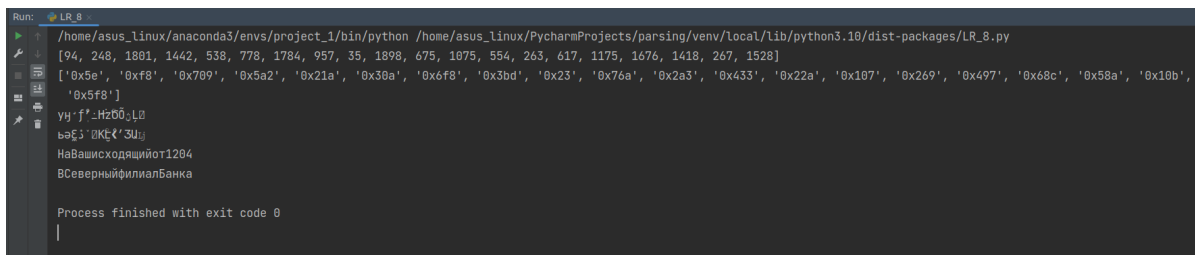
```

```

k = key(P_1)
print(k)
print(hex_key(k))
c_1 = encod_and_decod(P_1, k)
c_2 = encod_and_decod(P_2, k)
print(c_1)
print(c_2)
print(encod_and_decod(c_1, k))
print(encod_and_decod(c_2, k))

```

3.2. Результат работы



```

Run: LR_8
/home/asus_linux/anaconda3/envs/project_1/bin/python /home/asus_linux/PycharmProjects/parsing/venv/local/lib/python3.10/dist-packages/LR_8.py
[94, 248, 1801, 1442, 538, 778, 1784, 957, 35, 1898, 675, 1875, 554, 263, 617, 1175, 1676, 1418, 267, 1528]
['0x5e', '0xf8', '0x709', '0x5a2', '0x21a', '0x30a', '0x6f8', '0x3bd', '0x23', '0x76a', '0x2a3', '0x433', '0x22a', '0x107', '0x269', '0x497', '0x68c', '0x58a', '0x10b',
'0x5f8']
уу'f'~H200,LB
ьәҫ' 0Кt'3U
НаВашисходящийот1204
ВСеверныйфилиалБанка

Process finished with exit code 0

```

4. Выводы

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