

# Лабораторная работа №8

---

Бешкуров Михаил

17.12.2021

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

---

- Криптография - наука о методах шифрования. Умение шифровать различные исходные тексты одним ключом является необходимым для дальнейшего знакомства с криптографией.

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

- Написать программу, которая должна определять вид шифротекстов при известных открытых текстах и при известном ключе.
- Также эта программа должна определить вид одного из текстов, зная вид другого открытого текста и зашифрованный вид обоих текстов (т.е. не нужно использовать ключ при дешифровке).

- Написал функцию шифрования, которая определяет вид шифротекста при известном ключе и известных открытых текстах “НаВашисходящийот1204” и “ВСеверныйфилиалБанка”. Ниже представлены функция, шифрующая данные (рис - @fig:001), а также работа данной функции (рис - @fig:002).

```

In [1]: import numpy as np

In [2]: def encryption(text1, text2):
    print("Открытый 1ый текст: ", text1)
    # Задам массив из символов открытого 1го текста в шестнадцатеричном представлении:
    text_array1 = []
    for i in text1:
        text_array1.append(i.encode("cp1251").hex())
    print("\nОткрытый 1ый текст в шестнадцатеричном представлении: ", *text_array1)

    print("\nОткрытый 2ой текст: ", text2)
    # Задам массив из символов открытого 2го текста в шестнадцатеричном представлении:
    text_array2 = []
    for i in text2:
        text_array2.append(i.encode("cp1251").hex())
    print("\nОткрытый 2ой текст в шестнадцатеричном представлении: ", *text_array2)

    # Задам случайно сгенерированный ключ в шестнадцатеричном представлении:
    key_dec = np.random.randint(0, 255, len(text1))
    key_hex = [hex(i)[2:] for i in key_dec]
    print("\nКлюч в шестнадцатеричном представлении: ", *key_hex)

    # Задам зашифрованный 1ый текст в шестнадцатеричном представлении:
    crypt_text1 = []
    for i in range(len(text_array1)):
        crypt_text1.append("{:02x}".format(int(text_array1[i], 16) ^ int(key_hex[i], 16)))
    print("\nЗашифрованный 1ый текст в шестнадцатеричном представлении: ", *crypt_text1)

    # Задам зашифрованный 2ой текст в шестнадцатеричном представлении:
    crypt_text2 = []
    for i in range(len(text_array2)):
        crypt_text2.append("{:02x}".format(int(text_array2[i], 16) ^ int(key_hex[i], 16)))
    print("\nЗашифрованный 2ой текст в шестнадцатеричном представлении: ", *crypt_text2)

    # Задам зашифрованный 1ый текст в обычном представлении:
    final_text1 = bytearray.fromhex("".join(crypt_text1)).decode("cp1251")
    print("\nЗашифрованный 1ый текст: ", final_text1)

    # Задам зашифрованный 2ой текст в обычном представлении:
    final_text2 = bytearray.fromhex("".join(crypt_text2)).decode("cp1251")
    print("\nЗашифрованный 2ой текст: ", final_text2)

    return key_hex, final_text1, final_text2

```

Рис. 1: Функция, шифрующая данные

```

In [4]: #Изначальные фразы:
p1 = "НаВашисходящий1204"
p2 = "ВСеверныйфилиалБанка"
key, res1, res2 = encryption(p1, p2)

Открытый 1ый текст: НаВашисходящий1204
Открытый 1ый текст в шестнадцатеричном представлении: cd e0 c2 e0 f8 e8 f1 f5 ee e4 ff f9 e8 e9 ee f2 31 32 30 34
Открытый 2ой текст: ВСеверныйфилиалБанка
Открытый 2ой текст в шестнадцатеричном представлении: c2 d1 e5 e2 e5 f0 ed fb e9 f4 e8 eb e8 e0 eb c1 e0 ed ea e0
Ключ в шестнадцатеричном представлении: d6 17 38 2e 32 e3 6d ec 95 f6 a2 43 bd e4 39 79 f9 28 5d d4
Зашифрованный 1ый текст в шестнадцатеричном представлении: 1b f7 fa ce ca 0b 9c 19 7b 12 5d ba 55 0d d7 8b c8 1a 6d
e0
Зашифрованный 2ой текст в шестнадцатеричном представлении: 14 c6 dd cc d7 13 80 17 7c 02 4a a8 55 04 d2 b8 19 c5 b7
34
Числовой 1ый текст: чЮКы[]eU
Зашифрованный 2ой текст: ЖЭММ)JÉUTEЕ·4

```

Рис. 2: Результат работы функции, шифрующей данные



- Написал функцию дешифровки, которая определяет вид одного из текстов, зная вид другого открытого текста и зашифрованный вид обоих текстов (т.е. не использует ключ). (рис - @fig:003). А также представил результаты работы программы (рис - @fig:004).

```
In [3]: def decryption(cr_text1, cr_text2, op_text1):
        print("\nЗашифрованный 1ый текст: ", cr_text1)
        print("\nЗашифрованный 2ой текст: ", cr_text2)
        print("\nОткрытый 1ый текст: ", op_text1)

        cr_text_hex1 = []
        for i in cr_text1:
            cr_text_hex1.append(i.encode("cp1251").hex())
        print("\nЗашифрованный 1ый текст в 16ом представлении: ", *cr_text_hex1)

        cr_text_hex2 = []
        for i in cr_text2:
            cr_text_hex2.append(i.encode("cp1251").hex())
        print("\nЗашифрованный 2ой текст в 16ом представлении: ", *cr_text_hex2)

        op_text_hex1 = []
        for i in op_text1:
            op_text_hex1.append(i.encode("cp1251").hex())
        print("\nОткрытый 1ый текст в 16ом представлении: ", *op_text_hex1)

        cr1_cr2 = []
        op_text_hex2 = []
        for i in range(len(op_text1)):
            cr1_cr2.append("{}:02x".format(int(cr_text_hex1[i],16) ^ int(cr_text_hex2[i],16)))
            op_text_hex2.append("{}:02x".format(int(cr1_cr2[i], 16) ^ int(op_text_hex1[i], 16)))

        print("\nОткрытый 2ой текст в 16ом представлении: ", *op_text_hex2)
        op_text2 = bytearray.fromhex("".join(op_text_hex2)).decode("cp1251")
        print("\nОткрытый 2ой текст: ", op_text2)
        return op_text2
```

Рис. 3: Функция, дешифрующая данные

```

In [5]: text2 = decryption(res1, res2, p1)
print("\nОткрытый 2ой текст: ", text2)

Ч:Ипоаванный 1ый текст: чьОКы{}еU
Зашифрованный 2ой текст: ЖЭМФ}JËUтЕ-4
Открытый 1ый текст: НаВависходящийот1284

Зашифрованный 1ый текст в 16ом представлении: 1b f7 fa ce ca 0b 9c 19 7b 12 5d ba 55 0d d7 8b c8 1a 6d e9
Зашифрованный 2ой текст в 16ом представлении: 14 c6 dd cc d7 13 80 17 7c 02 4a a8 55 04 d2 b8 19 c5 b7 34

Открытый 1ый текст в 16ом представлении: cd e0 c2 e0 f8 e8 f1 f5 ee e4 ff f9 e8 e9 ee f2 31 32 30 34
Открытый 2ой текст в 16ом представлении: c2 d1 e5 e2 e5 f0 ed fb e9 f4 e8 eb e8 e0 eb c1 e0 ed ea e0
Открытый 2ой текст: ВСеверныйфилиалБанка

In [6]: text1 = decryption(res2, res1, p2)
print("\nОткрытый 1ый текст: ", text1)

Зашифрованный 1ый текст: ЖЭМФ}JËUтЕ-4
Ч:Ипоаванный 2ой текст: чьОКы{}еU
Открытый 1ый текст: ВСеверныйфилиалБанка

Зашифрованный 1ый текст в 16ом представлении: 14 c6 dd cc d7 13 80 17 7c 02 4a a8 55 04 d2 b8 19 c5 b7 34
Зашифрованный 2ой текст в 16ом представлении: 1b f7 fa ce ca 0b 9c 19 7b 12 5d ba 55 0d d7 8b c8 1a 6d e9

Открытый 1ый текст в 16ом представлении: c2 d1 e5 e2 e5 f0 ed fb e9 f4 e8 eb e8 e0 eb c1 e0 ed ea e0
Открытый 2ой текст в 16ом представлении: cd e0 c2 e0 f8 e8 f1 f5 ee e4 ff f9 e8 e9 ee f2 31 32 30 34
Открытый 2ой текст: НаВависходящийот1284
Открытый 1ый текст: НаВависходящийот1284

```

Рис. 4: Результат работы функции, дешифрующей данные

Таким образом, я освоил на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом.