

Лабораторная работа №8

Карымшаков А.А.- студент группы НФИбд-03-18

17.02.2022

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).

```

Ввод [1]: import numpy as np

Ввод [18]: 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

```

Ввод [20]: Изначальные фразы:
p1 = "НаВависходящийот1204"
p2 = "ВСеверныйФилиалБанка"
key, res1, res2 = encryption(p1, p2)

Открытый 1ый текст: НаВависходящийот1204
Открытый 1ый текст в шестнадцатеричном представлении: cd e0 c2 e0 f8 e8 f1 f5 ee e4 ff f9 e8 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: 7d 4e 4d de a2 75 c0 6c 4c a9 38 d f1 6a 4a 88 dc df 3d e2
Зашифрованный 1ый текст в шестнадцатеричном представлении: b0 ae 8f 3e 5a 9d 31 99 a2 4d c7 f4 19 83 a4 7a ed ed 0d d6
Зашифрованный 2ой текст в шестнадцатеричном представлении: bf 9f a8 3c 47 85 2d 97 a5 5d d0 e6 19 8a a1 49 3c 32 d7 02
Зашифрованный 1ый текст: "0p2x1~jнзф0'нzn
Зашифрованный 2ой текст: iuE<g...-f]Рх0b9I<2ч

```

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).

```
Ввод [19]: def decryption(cr_text1, cr_text2, op_text1):
    print("\nзашифрованный 1ый текст: ", cr_text1)
    print("\nзашифрованный 2ой текст: ", cr_text2)
    print("Открытый 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("Открытый 2ой текст в 16ом представлении: ", *op_text_hex2)
    op_text2 = bytearray.fromhex("".join(op_text_hex2)).decode("cp1251")
    print("Открытый 2ой текст: ", op_text2)
    return op_text2
```

Рис. 3: Функция, дешифрующая данные

```
Ввод [23]: text1 = decryption(res1, res2, p1)
print("\nОткрытый 2ой текст: ", text2)

Цифровой 1ый текст: 0x2011000000000000
Цифровой 2ой текст: 0x2011000000000000
Открытый 1ый текст: Вашисходящий1204

Цифровой 1ый текст в 16ом представлении: b0 ae 8f 3e 5a 9d 31 99 a2 4d c7 f4 19 83 a4 7a ed ed 0d
Цифровой 2ой текст в 16ом представлении: bf 9f a8 3c 47 85 2d 97 a5 5d d0 e6 19 8a a1 49 3c 32 d7 02

Открытый 1ый текст в 16ом представлении: cd e0 c2 e0 f8 e8 f1 f5 ee e4 ff f9 e8 ee e2 f1 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

Открытый 2ой текст: ВСеверныйФилиалБанка

Ввод [24]: text1 = decryption(res2, res1, p2)
print("\nОткрытый 1ый текст: ", text1)

Цифровой 1ый текст: 0x2011000000000000
Цифровой 2ой текст: 0x2011000000000000
Открытый 1ый текст: ВСеверныйФилиалБанка

Цифровой 1ый текст в 16ом представлении: bf 9f a8 3c 47 85 2d 97 a5 5d d0 e6 19 8a a1 49 3c 32 d7 02
Цифровой 2ой текст в 16ом представлении: b0 ae 8f 3e 5a 9d 31 99 a2 4d c7 f4 19 83 a4 7a ed ed 0d 0e

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 ee e2 f1 32 30 34
Открытый 2ой текст: Вашисходящий1204

Открытый 1ый текст: Вашисходящий1204
```

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.