Лабораторная работа №8

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

Дмитрий Константинович Федотов

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

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

Ответить на вопросы.

2 Выполнение лабораторной работы

1. Написал функцию шифрования, которая определяет вид шифротекста при известном ключе и известных открытых текстах (рис - @fig:001), а также работа данной функции (рис - @fig:002).

```
def func(text1, text2):
    print("Открытый текст 1: ", text1)
    text_ar1 = []
    for i in text1:
        text_ar1.append(i.encode("cp1251").hex())
    print("Текст 1 в шеснадцатеричном исполнении: ", *text_ar1)

    print("Открытый текст 2: ", text2)
    text_ar2 = []
    for i in text2:
        text_ar2.append(i.encode("cp1251").hex())
    print("Текст 2 в шеснадцатеричном исполнении: ", *text_ar2)

    key = np.random.randint(0, 255, len(text1))
    key_hex = [hex(i)[2:] for i in key]
    print("ключ в шеснацатеричнов виде: ", *key_hex)

    text1_crypt = []
    for i in range(len(text_ar1)):
        text1_crypt.append("{:02x}".format(int(text_ar1[i], 16)^ int(key_hex
    print("Зашифрованный текст 1 в шеснадцатеричной: ", *text1_crypt)

    text2_crypt = []
    for i in range(len(text_ar2)):
        text2_crypt.append("{:02x}".format(int(text_ar2[i], 16)^ int(key_hex
    print("Зашифрованный текст 2 в шеснадцатеричной: ", *text2_crypt)

    text1_ret = bytearray.fromhex("".join(text1_crypt)).decode("cp1251")
    print("Зашифрованный текст 1: ", text1_ret)

    text2_ret = bytearray.fromhex("".join(text2_crypt)).decode("cp1251")
    print("Зашифрованный текст 2: ", text2_ret)

    return key_hex, text1_ret, text2_ret
```

Рис. 2.1: Функция, шифрующая данные

```
]: start_text1 = "НаВашисходящийот1204"
   start_text2 = "ВСеверныйфилиалБанка"
   crypt_key, res1, res2 = func(start_text1, start_text2)

   Открытый текст 1:  НаВашисходящийот1204
   Текст 1 в шеснадцатеричном исполнении:  cd e0 c2 e0 f8 e8 f1 f5 ee e4 ff f9
   e8 e9 ee f2 31 32 30 34
   Открытый текст 2:  ВСеверныйфилиалБанка
   Текст 2 в шеснадцатеричном исполнении:  c2 d1 e5 e2 e5 f0 ed fb e9 f4 e8 eb
   e8 e0 eb c1 e0 ed ea e0
   ключ в шеснацатеричнов виде:  69 93 fe eb b2 60 22 1b 39 ed 20 20 bf 79 3a
   83 f7 72 d1 2
   Зашифрованный текст 1 в шеснадцатеричной:  a4 73 3c 0b 4a 88 d3 ee d7 09 df
   d9 57 90 d4 71 c6 40 e1 36
   Зашифрованный текст 2 в шеснадцатеричной:  ab 42 1b 09 57 90 cf e0 d0 19 c8
   cb 57 99 d1 42 17 9f 3b e2
   Зашифрованный текст 1:  ¤s<♂J€YoЧ       ЯЩЊЋфЖ@б6
   Зашифрованный текст 2:  «B←     WЋПаР├ИЛЊ™СВ┤ц;в
```

Рис. 2.2: Результат работы функции

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

```
def func2(text1_cr, text2_cr, text1):
    print("Открытый текст 1: ", text1)
    print("Зашифрованный текст 1: ", text1_cr)
    print("Зашифрованный текст 2: ", text2_cr)
    text1_cr_hex = []
    for i in text1_cr:
        text1_cr_hex.append(i.encode("cp1251").hex())
    print("зашифрованный текст 1 в шеснадцптеричной: ", *text1_cr_hex)

    text2_cr_hex = []
    for i in text2_cr:
        text2_cr_hex.append(i.encode("cp1251").hex())
    print("зашифрованный текст 2 в шеснадцптеричной: ", *text2_cr_hex)

    text1_hex = []
    for i in text1:
        text1_hex.append(i.encode("cp1251").hex())
    print("открытый текст 1 в шеснадцптеричной: ", *text1_hex)

    cr = []
    text2_hex = []
    for i in range(len(text1)):
        cr.append("{:02x}".format(int(text1_cr_hex[i],16) ^ int(text2_cr_hex
        text2_hex.append("{:02x}".format(int(cr[i],16) ^ int(text1_hex[i],16
    print("открытый текст 2 в шеснадцптеричной: ", *text2_hex)
    text2 = bytearray.fromhex("".join(text2_hex)).decode("cp1251")
    print("Открытый текст 2: ", text2)
    return text2
```

Рис. 2.3: Функция, дешифрующая данные

```
text2_ret = func2(res1, res2, start_text1)
print("Открытый текст 2:", text2_ret)
```

```
Открытый текст 1:  НаВашисходящийот1204
Зашифрованный текст 1:  ¤s<♂J€УоЧ       ЯЩЩЋфqЖ@б6
Зашифрованный текст 2:  «В←     ЩЋПаР├ИЛЩ™СВ┤u;в
зашифрованный текст 1 в шеснадцптеричной:  a4 73 3c 0b 4a 88 d3 ee d7 09 df
d9 57 90 d4 71 c6 40 e1 36
зашифрованный текст 2 в шеснадцптеричной:  ab 42 1b 09 57 90 cf e0 d0 19 c8
cb 57 99 d1 42 17 9f 3b e2
открытый текст 1 в шеснадцптеричной:  cd e0 c2 e0 f8 e8 f1 f5 ee e4 ff f9 e
8 e9 ee f2 31 32 30 34
открытый текст 2 в шеснадцптеричной:  c2 d1 e5 e2 e5 f0 ed fb e9 f4 e8 eb e
8 e0 eb c1 e0 ed ea e0
Открытый текст 2:  ВСеверныйфилиалБанка
Открытый текст 2: ВСеверныйфилиалБанка
```

Рис. 2.4: Результат работы функции

3 Выводы

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

4 Ответы на контрольные вопросы

1. $C_1 \oplus C_2 \oplus + P_1 = P_1 \oplus P_2 \oplus + P_1 = P_2$, где C_1 и C_2 - шифротексты. Т.е. ключ в данной формуле не используется.
2. При повторном использовании ключа при шифровании текста получим исходное сообщение.

3. с помощью формулы:

$$C_1 = P_1 \oplus + K$$

$$C_2 = P_2 \oplus + K,$$

   где C_i - шифротексты, P_i - открытые тексты, K - единый ключ шифровки

4. Недостатки шифрования одним ключом двух открытых текстов: Зная одно сообщение и два шифротекста, можно расшифровать без ключа.
   Зная шаблон сообщений, можно определить те символы сообщения P_2, которые находятся на позициях известного шаблона сообщения P_1.

5. Преимущества шифрования одним ключом двух открытых текстов:
   Процесс шифровки и дешифровки становится проще. Между двумя компьютерами, удобнее пользоваться одним ключом.