Лабораторная работа №8

Василий О. Худицкий

РУДН, 29 октября 2022, Москва, Россия

Цель лабораторной работы

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

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.

Выполнение лабораторной работы

Определение вида шифротекстов C_1 и C_2

```
def encryption(text1, text2):
        print("Открытый текст 1: ", text1)
        new_text1 = []
        for i in text1:
            new_text1.append(i.encode("cp1251").hex())
        print("\nОткрытый текст 1 в 16-ой системе: ", new_text1)
        print("\nОткрытый текст 2: ", text2)
        new_text2 = []
        for i in text2:
            new_text2.append(i.encode("cp1251").hex())
        print("\nОткрытый текст 2 в 16-ой системе: ", new_text2)
        r = np.random.randint(0, 255, len(text1))
        key = [hex(i)[2:] for i in r]
        new_key = []
        for i in key:
                new_key.append(i.encode("cp1251").hex().upper())
        print("\nКлюч в 16-ой системе: ", key)
        xor_text1 = []
        for i in range(len(new_text1)):
            xor_text1.append("{:02x}".format(int(key[i], 16) ^ int(new_text1[i], 16)))
        print("\nШифротекст 1 в 16-ой системе: ", xor_text1)
        c1 = bytearray.fromhex("".join(xor_text1)).decode("cp1251")
        print("\nШифротекст 1: ", c1)
        xor_text2 = []
        for i in range(len(new_text2)):
            xor_text2.append("{:02x}".format(int(key[i], 16) ^ int(new_text2[i], 16)))
        print("\nШифротекст 2 в 16-ой системе: ", xor_text2)
        c2 = bytearray.fromhex("".join(xor_text2)).decode("cp1251")
        print("\nШифротекст 2: ", c2)
        return key, xor_text1, c1, xor_text2, c2
```

Результат вызова первой функции

```
[4]  k, t1, c1, t2, c2 = encryption(p1, p2)

     Открытый текст 1:  С Новым Годом, друзья!

     Открытый текст 1 в 16-ой системе:  ['d1', '20', 'cd', 'ee', 'e2', 'fb', 'ec', '20', 'c3', 'ee', 'e4', 'ee', 'ec', '2c', '20', 'e4', 'f0', 'f3', 'e7', 'fc', 'ff', '21']

     Открытый текст 2:  С Новым Годом, коллеги

     Открытый текст 2 в 16-ой системе:  ['d1', '20', 'cd', 'ee', 'e2', 'fb', 'ec', '20', 'c3', 'ee', 'e4', 'ee', 'ec', '2c', '20', 'ea', 'ee', 'eb', 'eb', 'e5', 'e3', 'e8']

     Ключ в 16-ой системе:  ['a1', '5f', '34', '47', '9d', '3f', '78', '97', '33', '7', '94', '27', '2b', '79', '45', 'e3', 'b6', '89', '5d', 'c1', '23', '1f']

     Шифротекст 1 в 16-ой системе:  ['70', '7f', 'f9', 'a9', '7f', 'c4', '94', 'b7', 'f0', 'e9', '70', 'c9', 'c7', '55', '65', '07', '46', '7a', 'ba', '3d', 'dc', '3e']

     Шифротекст 1:  р щ© Д”·рйрЙЗUe▯Fzє=b›

     Шифротекст 2 в 16-ой системе:  ['70', '7f', 'f9', 'a9', '7f', 'c4', '94', 'b7', 'f0', 'e9', '70', 'c9', 'c7', '55', '65', '09', '58', '62', 'b6', '24', 'c0', 'f7']

     Шифротекст 2:  р щ© Д”·рйрЙЗUe  Xb¶$Ач
```

Рис.1 Вывод функции encryption

Определение способа, при котором злоумышленник может прочитать оба текста, не зная ключа и не стремясь его определить

```
def decryption(c1, c2, p1):
        print("Шифротекст 1: ", c1)
        new_c1 = []
        for i in c1:
            new_c1.append(i.encode("cp1251").hex())
        print("\nШифротекст 1 в 16-ой системе: ", new_c1)
        print("\nШифротекст 2: ", c2)
        new_c2 = []
        for i in c2:
            new_c2.append(i.encode("cp1251").hex())
        print("\nШифротекст 2 в 16-ой системе: ", new_c2)
        print("\nОткрытый текст 1: ", p1)
        new_p1 = []
        for i in p1:
            new_p1.append(i.encode("cp1251").hex())
        print("\nОткрытый текст 1 в 16-ой системе: ", new_p1)

        print("\nНахождение второй открытый текст...")
        xor_tmp = []
        sp2 = []
        for i in range(len(p1)):
            xor_tmp.append("{:02x}".format(int(new_c1[i], 16) ^ int(new_c2[i], 16)))
            sp2.append("{:02x}".format(int(xor_tmp[i], 16) ^ int(new_p1[i], 16)))
        print("\nОткрытый текст 2 в 16-ой системе: ", sp2)
        p2 = bytearray.fromhex("".join(sp2)).decode("cp1251")
        print("\nОткрытый текст 2: ", p2)
        return p2, sp2
```

Результат вызова второй функции

```
[6]  s3 = decryption(c1, c2, p1)

     Шифротекст 1:  р щ© Д”·рйрЙЗUe▣Fzє=Ь›

     Шифротекст 1 в 16-ой системе:  ['70', '7f', 'f9', 'a9', '7f', 'c4', '94', 'b7', 'f0', 'e9', '70', 'c9', 'c7', '55', '65', '07', '46', '7a', 'ba', '3d', 'dc', '3e']

     Шифротекст 2:  р щ© Д”·рйрЙЗUe  Xb¶$Ач

     Шифротекст 2 в 16-ой системе:  ['70', '7f', 'f9', 'a9', '7f', 'c4', '94', 'b7', 'f0', 'e9', '70', 'c9', 'c7', '55', '65', '09', '58', '62', 'b6', '24', 'c0', 'f7']

     Открытый текст 1:  С Новым Годом, друзья!

     Открытый текст 1 в 16-ой системе:  ['d1', '20', 'cd', 'ee', 'e2', 'fb', 'ec', '20', 'c3', 'ee', 'e4', 'ee', 'ec', '2c', '20', 'e4', 'f0', 'f3', 'e7', 'fc', 'ff', '21']

     Нахождение второй открытый текст...

     Открытый текст 2 в 16-ой системе:  ['d1', '20', 'cd', 'ee', 'e2', 'fb', 'ec', '20', 'c3', 'ee', 'e4', 'ee', 'ec', '2c', '20', 'ea', 'ee', 'eb', 'eb', 'e5', 'e3', 'e8']

     Открытый текст 2:  С Новым Годом, коллеги
```

Рис.2 Вывод функции decryption

Выводы

В ходе выполнения данной лабораторной работы я освоил на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом.