Отчёт по лабораторной работе 2

Шифры перестановки

Каримов Зуфар Исматович НФИ-01-22

Содержание

List of Tables

List of Figures

1 Цель работы

Реализация маршрутного шифра, решетчатого шифра и таблицы Виженера.

2 Теоретические сведения

1- Маршрутное шифрование

Этот способ шифрования изобрел выдающийся французский математик и криптограф Франсуа Виет (1540-1603).

Пусть m и n – некоторые натуральные (т.е. целые положительные) числа, каждое больше 1. Открытый текст последовательно разбивается на части (блоки) с длиной, равной произведению mn (если в последнем блоке не хватает букв, можно дописать до нужной длины произвольный их набор). Блок вписывается построчно в таблицу размерности m×n (т.е. m строк и n столбцов). Криптограмма получается выписыванием букв из таблицы в соответствии с некоторым маршрутом. Этот маршрут вместе с числами m и n составляет ключ шифра.

Чаще всего буквы выписывают по столбцам, которые упорядочиваются в соответствии с паролем: под таблицей подписывается слово, состоящее из n неповторяющихся букв, и столбцы таблицы нумеруются по алфавитному порядку букв пароля. Например, для шифрования открытого текста, выражающего один из главных принципов криптологии: нельзя недооценивать противника, добавим к его 29 буквам еще одну, скажем а, возьмем m=5, n=6, впишем текст в таблицу 5×6 и выберем в качестве пароля слово п а р о л ь:

нельзя недооц ениват ьпроти вникаа пароль

Выписывая теперь буквы по столбцам в соответствии с алфавитным порядком букв в пароле, получаем следующую криптограмму: ЕЕНПНЗОАТАЬОВОКННЕЬ-ВЛДИРИЯЦТИА (истинные пробелы в криптографии не выставляются).

Выберите другой пароль и посмотрите, как изменится криптограмма.

Рассмотренный способ шифрования (столбцовая перестановка) в годы первой мировой войны использовала легендарная немецкая шпионка Мата Хари.

2- Шифрование с помощью решеток

Этот способ шифрования предложил в 1881 году австрийский криптограф Эдуард Флейснер. Выбирается натуральное число k > 1, и квадрат размерности k×k построчно заполняется числами 1, 2, …, k. Для примера возьмем k = 2.

Квадрат поворачивается по часовой стрелке на 90° и размещается вплотную к предыдущему квадрату. Аналогичные действия совершаются еще два раза, так чтобы в результате из четырех малых квадратов образовался один большой с длиной стороны 2k.

Далее из большого квадрата вырезаются клетки с числами от 1 до k2, для каждого числа одна клетка. Процесс шифрования происходит следующим образом. Сделанная решетка (квадрат с прорезями) накладывается на чистый квадрат 2k×2k и в прорези по строчкам (т.е. слева направо и сверху вниз) вписываются первые буквы открытого текста. Затем решетка поворачивается на 90° по часовой стрелке и накладывается на частично заполненный квадрат, вписывание продолжается.

После третьего поворота, наложения и вписывания все клетки квадрата будут заполнены. Правило выбора прорезей гарантирует, что при заполнении квадрата буква на букву никогда не попадет. Из заполненного квадрата буквы можно выписать по столбцам, выбрав подходящий пароль. Например, с использованием изображенной выше решетки и пароля ш и ф р открытый текст договор подписали переводится в криптограмму за пять шагов.

Итоговая криптограмма: ОВОРДЛГПАПИОСДОИ.[1]

3- Шифр Виженера

Шифр Виженера (фр. Chiffre de Vigenère) — метод полиалфавитного шифрования буквенного текста с использованием ключевого слова.

Этот метод является простой формой многоалфавитной замены. Шифр Виженера изобретался многократно. Впервые этот метод описал Джован Баттиста

Беллазо (итал. Giovan Battista Bellaso) в книге La cifra del. Sig. Giovan Battista Bellaso в 1553 году, однако в XIX веке получил имя Блеза Виженера, французского дипломата. Метод прост для понимания и реализации, он является недоступным для простых методов криптоанализа.

В шифре Цезаря каждая буква алфавита сдвигается на несколько строк; например в шифре Цезаря при сдвиге +3, A стало бы D, B стало бы E и так далее. Шифр Виженера состоит из последовательности нескольких шифров Цезаря с различными значениями сдвига. Для зашифровывания может использоваться таблица алфавитов, называемая tabula recta или квадрат (таблица) Виженера. Применительно к латинскому алфавиту таблица Виженера составляется из строк по 26 символов, причём каждая следующая строка сдвигается на несколько позиций. Таким образом, в таблице получается 26 различных шифров Цезаря. На каждом этапе шифрования используются различные алфавиты, выбираемые в зависимости от символа ключевого слова. Например, предположим, что исходный текст имеет вид:

ATTACKATDAWN

Человек, посылающий сообщение, записывает ключевое слово («LEMON») циклически до тех пор, пока его длина не будет соответствовать длине исходного текста:

LEMONLEMONLE

Первый символ исходного текста A зашифрован последовательностью L, которая является первым символом ключа. Первый символ L шифрованного текста находится на пересечении строки L и столбца A в таблице Виженера. Точно так же для второго символа исходного текста используется второй символ ключа; то есть второй символ шифрованного текста X получается на пересечении строки E и столбца T. Остальная часть исходного текста шифруется подобным способом.

Исходный текст: ATTACKATDAWN

Ключ: LEMONLEMONLE

Зашифрованный текст: LXFOPVEFRNHR [2]

3 Выполнение лабораторной работы

1. Написал функцию для маршрутного шифрование для текста. (рис. 3.1) (рис. 3.2) (рис. 3.3)

Написал функции для определения индекс буквы в нашем ключе, а затем пополнил таблицу с сообщением. в конце распечатал шифрование в порядке индекса ключа

```python
def encryptMessage(msg, key):
    cipher = ""

    msg = msg.replace(' ', '')

    # берём длину текста
    msg_len = int(len(msg))

    # создаём список букв этого текста
    msg_lst = list(msg)

    # сортируем буквы ключа по алфавиту
    key_lst = sorted(list(key))

    # считаем количество столбцов
    col = len(key)

    # считаем количество строк
    if msg_len % col == 0:
        row = int(msg_len / col)
    else:
        row = int(msg_len // col) + 1

    # если наша таблица не полная, добовляем доп букву
    fill = int((row * col) - msg_len)
    msg_lst.extend('a' * fill)

    # создадим матрицу нужного размера для шифрования
    matrix = [msg_lst[i: i + col] for i in range(0, len(msg_lst), col)]

    for i in range(col):
        # так как до этого мы сортировали в алфавитном порядке,
        # теперь нам надо найти эти буквы в исходном ключе
        curr_idx = key.index(key_lst[i])
        # и соединить это всё в одну строку
        cipher += ''.join([row[curr_idx] for row in matrix])

    return cipher
```

Figure 3.1: Функция для кодирования текста шифром Маршрутного шифрования

```python
def decryptMessage(cipher, key):
    msg = ""

    # берём длину текста
    msg_len = int(len(cipher))

    # создаём список букв этого текста
    msg_lst = list(cipher)

    # считаем количество столбцов
    col = len(key)

    # считаем количество столбцов
    row = int(msg_len / col)

    # сортируем буквы ключа
    key_lst = sorted(list(key))

    # создаём пустую матрицу размера, который мы высчитали
    dec_cipher = []
    for _ in range(row):
        dec_cipher += [[None] * col]

    msg_indx = 0

    for i in range(col):
        # так как до этого мы сортировали в алфавитном порядке,
        # теперь нам надо найти эти буквы в исходном ключе
        curr_idx = key.index(key_lst[i])

        for j in range(row):
            dec_cipher[j][curr_idx] = msg_lst[msg_indx]
            msg_indx += 1

    msg = ''.join(sum(dec_cipher, []))

    return msg
```

Figure 3.2: Функция для кодирования текста шифром Маршрутного расшифрования

```python
while True:
    msg = input("Введите сообщение для шифрования:\n")
    if (False in [x in a for x in msg]):
        continue
    else:
        break

while True:
    key = input("\nВведите ключ\n")
    if len(set(key)) != len(key):
        continue
    else:
        break

print("\nВаше зашифрованное сообщение: " + encryptMessage(msg, key))
```

Figure 3.3: Блок кода для вывода результат в соответствии с шифром Маршрутного шифрования

Получил результат. (рис. 3.4) (рис. 3.5)

```
while True:
    msg = input("Введите сообщение для шифрования:\n")
    if (False in [x in a for x in msg]):
        continue
    else:
        break

while True:
    key = input("\nВведите ключ\n")
    if len(set(key)) != len(key):
        continue
    else:
        break

print("\nВаше зашифрованное сообщение: " + encryptMessage(msg, key))
```

```
Введите сообщение для шифрования:
privet

Введите ключ
privet

Ваше зашифрованное сообщение: eiprtv
```

Figure 3.4: Получение шифрования текста методом Маршрутного шифрования

```
msg = input("Введите сообщение для расшифрования: ")
key = input("\nВведите ключ: ")

print("\nВаше расшифрованное сообщение: " + decryptMessage(msg,key))
```

```
Введите сообщение для расшифрования: privet

Введите ключ: privet

Ваше расшифрованное сообщение: ivrtpe
```

Figure 3.5: Получение шифрования текста методом Маршрутного расшифрования

3. Написал функцию vingere для шифрования Виженера. (рис. 3.6)

Сначала написал алфавит в виде списки.

Потом определил индекс каждой буквы в сообщении, аналогично ключу.

Как определил позицию, сложил на него позиции ключа. потом распечатал зашифрованный текст.

```python
def genKey(msg, key):
    key = list(key)
    if len(msg) == len(key):
        return(key)
    else:
        for i in range(len(msg) -
                       len(key)):
            key.append(key[i % len(key)])
    return("" . join(key))
```

```python
# шифрование
def vig(msg, key):
    cipher_text = []
    for i in range(len(msg)):
        x = (ord(msg[i]) + ord(key[i])) % 26
        x += ord('A')
        cipher_text.append(chr(x))
    return("" . join(cipher_text))
```

```python
def cipherText(string, key):
    cipher_text = []
    for i in range(len(string)):
        x = (ord(string[i]) +
             ord(key[i])) % 26
        x += ord('A')
        cipher_text.append(chr(x))
    return("" . join(cipher_text))
```

```python
# расшифровка
def unvig(cipher_text, key):
    orig_text = []
    #key.replace(' ','')
    for i in range(len(cipher_text)):
        x = (ord(cipher_text[i]) - ord(key[i]) + 26) % 26
        x += ord('A')
        orig_text.append(chr(x))
    return("" . join(orig_text))
```

Figure 3.6: Функция для кодирования текста шифром Фиженера

Получил результат. (рис. 3.7) (рис. 3.8)

```
while True:
    msg = input("Введите сообщение для шифрования:\n")
    if (False in [x in a1 for x in msg]):
        continue
    else:
        msg = msg.upper()
        break

while True:
    key = input("\nВведите ключ:\n")
    if (False in [x in a1 for x in msg]):
        continue
    else:
        key = key.upper()
        break

keyg = genKey(msg,key)
print("\nВаше зашифрованное сообщение: " + vig(msg, keyg))
```

```
Введите сообщение для шифрования:
privet

Введите ключ:
privet

Ваше зашифрованное сообщение: EIQQIM
```

Figure 3.7: Получение шифрования текста методом Фиженера

```
while True:
    msg = input("Введите сообщение для расшифрования: ")
    if (False in [x in a1 for x in msg]):
        continue
    else:
        msg = msg.upper()
        break

while True:
    key = input("\nВведите ключ: ")
    if (False in [x in a1 for x in msg]):
        continue
    else:
        key = key.upper()
        break

keyg = genKey(msg,key)
print("\nВаше расшифрованное сообщение: " + unvig(msg,keyg))
```

```
Введите сообщение для расшифрования: hello

Введите ключ: hello

Ваше расшифрованное сообщение: AAAAA
```

Figure 3.8: Получение расшифрования текста методом Фиженера

4 Выводы

Реализовал шифрование с помощью решеток, маршрутное шифрование и шифр Виженера

5 Список литературы

1. Перестановочные шифры.— URL: https://it.rfei.ru/course/$_{k017/}$7mdCpor7/~c5kOtaHY informatika/shifry_prostoy_zameny.
2. Шифр Виженера. — URL: https://www.sites.google.com/site/kriptografics/sifr-vizenera/.