лабораторной работе №2

Шифрыперестановки

Хамза хуссен

Цели работы

Изучение и анализ трёх базовых технологий шифрования:
- Шифрование методом маршрутной перестановки символов.
- Шифрование с использованием решёток (например, решётки Кардано).
- Шифр Виженера как одной из форм полиалфавитного шифрования.

Основные положения

Шифр маршрутной перестановки

Широкое применение находят шифры перестановки, основанные на геометрических фигурах. Алгоритм заключается в том, что исходный текст записывается в фигуру по одному маршруту, а затем считывается по другому. Такой метод называется маршрутной перестановкой.

Например, сообщение можно вписать в прямоугольную таблицу, используя маршрут: по горизонтали, начиная с левого верхнего угла, попеременно слева направо и справа налево. Для чтения зашифрованного текста применяется другой маршрут: по вертикали, начиная с правого верхнего угла, попеременно сверху вниз и снизу вверх.

Шифр Кардано

Решётка Кардано — это инструмент для кодирования и декодирования, представляющий собой прямоугольную (часто квадратную) карточку с вырезанными ячейками.

Карточка накладывается на носитель, и в прорези вписываются буквы сообщения. Затем таблица поворачивается (обычно вокруг вертикальной или горизонтальной оси, или на 90 градусов), и процесс повторяется. Таким образом, за четыре этапа заполняется вся таблица.

Для расшифровки необходимо наложить решётку на зашифрованный текст в правильной ориентации и считать символы в вырезанных ячейках. Этот метод был предложен Джероламо Кардано в 1550 году.

Шифр Виженера

Шифр Виженера — это метод полиалфавитного шифрования, использующий ключевое слово для сдвига символов исходного текста.

В отличие от шифра Цезаря, где все символы сдвигаются на одно и то же число позиций, в шифре Виженера применяется последовательность различных сдвигов, определяемых буквами ключа. Для шифрования используется таблица (tabula recta), содержащая 26 алфавитов, сдвинутых друг относительно друга.

Метод был впервые описан Джован Баттиста Беллазо в 1553 году, но стал известен под именем французского дипломата Блеза Виженера. Шифр устойчив к простым методам криптоанализа благодаря использованию нескольких алфавитов замены.

Выполнение работы

Реализация шифра маршрутной перестановки на языке Python

```
def marhsrutshifr():
    text = input("Введите текст: ").replace(' ', '')
    n = int(input("Введите число n: "))
    m = int(input("Введите число m: "))
    parol = input("Введите слово-пароль: ")
```

```
    lists = [['a' for _ in range(n)] for _ in range(m)]
    it = 0

    for i in range(m):
        for j in range(n):
            if it < len(text):
                lists[i][j] = text[it]
                it += 1

    lists.append(list(parol))

    spisok = sorted(lists[-1])
    result = ""

    for i in spisok:
        col_index = lists[-1].index(i)
        for j in range(m):
            result += lists[j][col_index]

    print(result)

marhsrutshifr()
```

```
marhsrutshifr()
```

```
Введите текст: Всем привет
Введите число n: 6
Введите число m: 4
Введите слово-пароль: Москва
Виаараааапааамтаасваааееаа
```

Реализация шифра решеткой на языке Python

```
import numpy as np

def rot90(matrix):
    return np.rot90(matrix).tolist()

def udalenie(largelist, inn, k):
    for i in range(k*2):
        for j in range(k*2):
            if largelist[i][j] == inn:
                largelist[i][j] = " "
```

```
                return

def cardangrille():
    k = int(input("Введите число k: "))
    s = 1
    lists = [[s + j + i*k for j in range(k)] for i in range(k)]

    lists1 = rot90(lists)
    lists2 = rot90(lists1)
    lists3 = rot90(lists2)

    largelist = [[1 for _ in range(2*k)] for _ in range(2*k)]

    for i in range(k):
        for j in range(k):
            largelist[i][j] = lists[i][j]
            largelist[i][j+k] = lists1[i][j]
            largelist[i+k][j+k] = lists2[i][j]
            largelist[i+k][j] = lists3[i][j]

    text = "договорподписали"
    largelist_a = [[" " for _ in range(2*k)] for _ in range(2*k)]

    li = list(range(1, k**2 + 1))

    for inn in li:
        udalenie(largelist, inn, k)

    for _ in range(4):
        for i in range(k*2):
            for j in range(k*2):
                if largelist[i][j] == largelist_a[i][j] and text:
                    largelist_a[i][j] = text[0]
                    text = text[1:]
        largelist = rot90(largelist)

    stri = input("Введите пароль: ")

    if len(stri) > k*2:
        stri = stri[:k*2]
    else:
        stri = stri.ljust(k*2, "z")

    largelist_a.append(list(stri))
```

```
    result = ""
    spisok = sorted(largelist_a[-1])

    for i in spisok:
        col_index = largelist_a[-1].index(i)
        for j in range(len(largelist_a)-1):
            if largelist_a[j][col_index] != " ":
                result += largelist_a[j][col_index]

    print(result.replace(" ", ""))

cardangrille()
```

```
Введите число k: 5
Введите пароль: Москва
ооооддвосопги
```

Реализация шифра Виженера на языке Python

```
import numpy as np

def rot90(matrix):
    return np.rot90(matrix).tolist()

def udalenie(largelist, inn, k):
    for i in range(k*2):
        for j in range(k*2):
            if largelist[i][j] == inn:
                largelist[i][j] = " "
                return

def cardangrille():
    k = int(input("Введите число k: "))
    s = 1
    lists = [[s + j + i*k for j in range(k)] for i in range(k)]

    lists1 = rot90(lists)
    lists2 = rot90(lists1)
    lists3 = rot90(lists2)

    largelist = [[1 for _ in range(2*k)] for _ in range(2*k)]
```

```
    for i in range(k):
        for j in range(k):
            largelist[i][j] = lists[i][j]
            largelist[i][j+k] = lists1[i][j]
            largelist[i+k][j+k] = lists2[i][j]
            largelist[i+k][j] = lists3[i][j]

    text = "договорподписали"
    largelist_a = [[" " for _ in range(2*k)] for _ in range(2*k)]

    li = list(range(1, k**2 + 1))

    for inn in li:
        udalenie(largelist, inn, k)

    for _ in range(4):
        for i in range(k*2):
            for j in range(k*2):
                if largelist[i][j] == largelist_a[i][j] and text:
                    largelist_a[i][j] = text[0]
                    text = text[1:]
        largelist = rot90(largelist)

    stri = input("Введите пароль: ")

    if len(stri) > k*2:
        stri = stri[:k*2]
    else:
        stri = stri.ljust(k*2, "z")

    largelist_a.append(list(stri))

    result = ""
    spisok = sorted(largelist_a[-1])

    for i in spisok:
        col_index = largelist_a[-1].index(i)
        for j in range(len(largelist_a)-1):
            if largelist_a[j][col_index] != " ":
                result += largelist_a[j][col_index]

    print(result.replace(" ", ""))

cardangrille()
```

```
Hello worldkey[107, 101, 121][72, 101, 108, 108, 111, 32, 119, 111, 114, 108,
100]Compare full encode {0: [72, 107], 1: [101, 101], 2: [108, 121], 3: [108,
107], 4: [111, 101], 5: [32, 121], 6: [119, 107], 7: [111, 101], 8: [114, 121],
9: [108, 107], 10: [100, 101]}
Шифр= 4KfXUcUlXJ
Deshifre= {0: [52, 107], 1: [75, 101], 2: [102, 121], 3: [88, 107], 4: [85, 101],
5: [26, 121], 6: [99, 107], 7: [85, 101], 8: [108, 121], 9: [88, 107], 10: [74,
101]}
Decode list= [72, 101, 108, 108, 111, 32, 119, 111, 114, 108, 100]
Word= Hello world
```

Из проведённой работы можно сделать следующие выводы:

1. Шифры перестановки представляют собой классические методы криптографии, основанные на изменении порядка символов исходного текста согласно определённому алгоритму.

2. Маршрутная перестановка позволяет наглядно продемонстрировать применение геометрических структур (таблиц) для шифрования текста.

3. Решётка Кардано иллюстрирует идею использования вращающихся шаблонов для многоэтапного заполнения и чтения зашифрованных данных.

4. Шифр Виженера является переходным звеном от моноалфавитных к полиалфавитным шифрам, что повышает стойкость к частотному анализу.