Отчёт по лабораторной работе №2

Дисциплина: Математические основы защиты информации и информационной безопасности

Полиенко Анастасия Николаевна, НПМмд-02-23

Содержание

Список иллюстраций

1 Цель работы

Изучить шифры перестановки.

2 Задание

1. Реализовать маршрутное шифрование.
2. Реализовать шифрование с помощью решёток.
3. Реализовать шифрование таблицей Виженера.

3 Теоретическое введение

Шифры перестановки преобразуют открытый текст в криптограмму путем перестановки его символов. Способ, каким при шифровании переставляются буквы открытого текста, и является ключом шифра. Важным требованием является равенство длин ключа и исходного текста.

Существует два широко распространенных метода перестановок:

1. **Маршрутное шифрование.**

Данный способ шифрования разработал французский математик Франсуа Виет. Открытый текст записывают в некоторую геометрическую фигуру (обычно прямоугольник) по некоторому пути, а затем, выписывая символы по другому пути, получают шифртекст. Пусть m и n - целые положительные числа, большие 1. Открытый текст разбивается на блоки равной длины, состоящие из числа символов, равному произведению mn. Если последний блок получится меньше остальных, то в него следует дописать требуемое количество произвольных символов. Составляется таблица размерности mn. Блоки вписывается построчно в таблицу. Криптограмма получается выписыванием букв из таблицы в соответствии с некоторым маршрутом. Ключом такой криптограммы является маршрут и числа m и n. Обычно буквы выписывают по столбцам, которые упорядочивают согласно паролю: внизу таблицы приписывается слово из неповторяющихся букв и столбцы нумеруются по алфавитному порядку букв пароля.

2. **Шифрование с помощью решеток.**

Данный способ шифрования предложил австрийский криптограф Эдуард Флейснер в 1881 году. Суть этого способа заключается в следующем. Выбирается натуральное число $k > 1$, строится квадрат размерности k и построчно заполняется числами $1, 2, \ldots, k^2$. Повернем его по часовой стрелке на 90° и присоединим к исходному квадрату справа. Проделаем еще дважды такую процедуру и припишем получившиеся квадраты снизу. Получился большой квадрат размерности $2k$.

Далее из большого квадрата вырезаются клетки, содержащие числа от 1 до k^2. В каждой клетке должно быть только одно число. Получается своего рода решето. Шифрование осуществляется следующим образом. Решето накладывается на чистый квадрат 2k × 2k и в прорези вписываются буквы исходного текста по порядку их следования. Когда заполнятся все прорези, решето поворачивается на 90° и вписывание букв продолжается. После третьего поворота все клетки большого квадрата окажутся заполненными. Подобрав подходящий пароль (число букв пароля должно равняться k^2 и они не должны повторяться), выпишем буквы по столбцам. Очередность столбцов определяется алфавитным порядком букв пароля.

Важно отметить, что число к подбирается в соответствии с количеством букв N исходного теста. В идеальном случае $k^2 = N$. Если такого равенства достичь невозможно, то можно либо дописать произвольную букву к последнему слову открытого текста, либо убрать ее.

3. **Таблица Виженера.**

В 1585 году французский криптограф Блез Виженер опубликовал свой метод шифрования в «Трактате о шифрах». Шифр считался нераскрываемым до 1863 года, когда австриец Фридрих Казиски взломал его.

Открытый текст разбивается на блоки длины n. Ключ представляет собой последовательность из n натуральных чисел: $a_1, \ldots, a_n$. Далее в каждом блоке первая буква циклически сдвигается вправо по алфавиту на a_1 позиций, вторая

буква - на a_2 позиций, последняя - на a_n позиций. Для лучшего запоминания в качестве ключа можно взять осмысленное слово, а алфавитные номера входящих в него букв использовать для осуществления сдвигов.

Более подробно см. в [1–6].

4 Выполнение лабораторной работы

1. Реализуем маршрутное шифрование (рис. 4.1).

Маршрутное шифрование

```
# ввод данных
message = input("Введите сообщение: ").replace(' ', '')
password = input("Введите пароль: ")
m = len(message)
n = len(password)
message += ((m % n)*n - m) * message[-1]
main_table = list()

#формирование таблицы
for i in range(m % n):
    main_table.append(list(message[(i*n): ((i+1)*n)]))

# Выбор порядка столбцов
sort_password = sorted(list(password))
list_index = list()
for i in sort_password:
    list_index.append(password.find(i))

# Шифрование сообщения
coded_message = ''
for i in list_index:
    for j in range(len(main_table)):
        coded_message += main_table[j][i]
print("Зашифрованное сообщение:", coded_message)
```

```
Введите сообщение: нельзя недооценивать противника
Введите пароль: пароль
Зашифрованное сообщение: еенпнзоатаьовокннеьвлдирияцтиа
```

Рис. 4.1: Код. Маршрутное шифрование

2. Реализуем шифрование с помощью решёток (рис. 4.2).

```
# ввод данных
k = int(input('Введите число k: '))
message = input("Введите сообщение: ").replace(' ', '')
password = input("Введите пароль длины k^2 без повторяющихся букв: ")
m = len(message)
message += (k**2 - m) * message[-1]

#формирование решётки
l = np.arange(k**2).reshape(k, k) + 1
table = np.hstack([np.vstack([l, np.rot90(l)]), np.vstack([np.rot90(l, k=3), np.rot90(l, k=2)])])
for i in range(k**2):
    r_index = np.random.randint(4)
    a = np.where(table == (i+1))[0][r_index]
    b = np.where(table == (i+1))[1][r_index]
    table[a][b] = 0

table1 = table; table2 = np.rot90(table, k=3); table3 = np.rot90(table, k=2); table4 = np.rot90(table)
index_1 = np.where(table1 == 0); index_2 = np.where(table2 == 0); index_3 = np.where(table3 == 0);
index_4 = np.where(table4 == 0)
index = np.hstack([index_1, index_2, index_3, index_4])
coded_table = np.empty((2*k, 2*k), dtype="object")

# Шифрование сообщения
for i in range(4):
    for j in range(k**2):
        coded_table[index[0][(k**2)*i + j]][index[1][(k**2)*i + j]] = message[(k**2)*i + j]
sort_password = sorted(list(password))
list_index = list()
for i in sort_password:
    list_index.append(password.find(i))
coded_message = ''
for i in list_index:
    for j in range(len(coded_table)):
        coded_message += coded_table[j][i]
print("Зашифрованное сообщение:", coded_message)
```

```
Введите число k: 2
Введите сообщение: договор подписали
Введите пароль длины k^2 без повторяющихся букв: шифр
Зашифрованное сообщение: олпрдигпаооосдви
```

Рис. 4.2: Код. Шифрование с помощью решёток

3. Реализуем шифрование таблицей Виженера (рис. 4.3).

Таблица Виженера

```
# ввод данных
alpha = 'абвгдеёжзийклмнопрстуфхцчшщъыьэюя'
message = input("Введите сообщение: ").replace(' ', '')
password = input("Введите пароль: ")

#уравнивание пароля с сообщением
i = 0
while len(message) != len(password):
    password += password[i]
    i += 1

# Шифрование сообщения
coded_message = ''
for i in range(m):
    ch1 = message[i]
    ch2 = password[i]
    ch3 = alpha[(alpha.index(ch1) + alpha.index(ch2)) % n]
    coded_message += ch3
print("Зашифрованное сообщение:",coded_message)
```

```
Введите сообщение: криптография серьёзная наука
Введите пароль: математика
Зашифрованное сообщение: чрыфяохщкфхядйэьшршалнтшча
```

Рис. 4.3: Код. Шифрование таблицей Виженера

5 Выводы

Изучила шифры перестановки на примере маршрутного шифрования, шифрования с помощью решёток и шифрования таблицей Виженера.

Список литературы

1. GNU Bash Manual [Электронный ресурс]. Free Software Foundation, 2016. URL: https://www.gnu.org/software/bash/manual/.
2. Newham C. Learning the bash Shell: Unix Shell Programming. O'Reilly Media, 2005. 354 с.
3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.
4. Robbins A. Bash Pocket Reference. O'Reilly Media, 2016. 156 с.
5. Таненбаум Э. Архитектура компьютера. 6-е изд. СПб.: Питер, 2013. 874 с.
6. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.