**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский Авиационный Институт»**

**(Национальный Исследовательский Университет)**

**Институт: №8 «Информационные технологии   
и прикладная математика»   
Кафедра: 806 «Вычислительная математика   
и программирование»**

Лабораторная работа № 3   
по курсу «Криптография»

Группа: М8О-307Б-21

Студент(ка): И. А. Лютоев

Преподаватель: А. В. Борисов

Оценка:

Дата: 23.04.2024

Москва, 2024

**ОГЛАВЛЕНИЕ**

[1](#_gjdgxs) Тема 3

[2](#_30j0zll) Задание 3

[3](#_1fob9te) Теория 4

[4](#_3znysh7) Ход лабораторной работы 6

[5](#_2et92p0) Выводы 7

# **Тема**

Сравнение текстов

# **Задание**

Сравнить 1) два осмысленных текста на естественном языке, 2) осмысленный текст и текст из случайных букв, 3) осмысленный текст и текст из случайных слов, 4) два текста из случайных букв, 5) два текста из случайных слов.

Считать процент совпадения букв в сравниваемых текстах – получить дробное значение от 0 до 1 как результат деления количества совпадений на общее число букв. Расписать подробно в отчёте алгоритм сравнения и приложить сравниваемые тексты в отчёте хотя бы для одного запуска по всем пяти случаям. Осознать какие значения получаются в этих пяти случаях.Привести соображения о том почему так происходит.

Длина сравниваемых текстов должна совпадать. Привести соображения о том какой длины текста должно быть достаточно для корректного сравнения.

# **Теория**

Индекс совпадений — один из методов криптоанализа шифра Виженера. Метод основывается на вычислении вероятности того, что два случайных элемента текста совпадут.

Рассмотрим текст, написанный на некотором языке. Алфавит данного языка будем полагать состоящим из m символов. Рассмотрим достаточно длинную строку x из n символов. Индексом совпадения называют вероятность совпадения двух произвольных символов в строке. Если fi — количество i-го символа алфавита в строке x, то индекс совпадения вычисляется по формуле:

Для английских текстов индекс совпадений равен примерно 7%, но для случайных текстов этот индекс равен , т.е. приблизительно 3%.

Алгоритм нахождения длины ключа. Разобьём текст на столбцы размера t. Если t кратно длине ключа, то каждые два элемента текста, отстоящие друг от друга на позиций, , зашифрованы одним и тем же алфавитом. А это означает, что каждая строка в выписанной выше таблице получена из открытого текста перестановкой. Если же t} не кратно длине ключа, то строки являются полиалфавитным шифром.

Ранее было показано, что индекс совпадений для перестановки открытого текста и для полиалфавитного шифра заметно отличается. Таким образом, перебирая различные значения t и вычисляя для каждого из них индекс совпадений, мы можем выделить те t, которые кратны длине ключа. Определить длину ключа по этим данным не составляет труда.

Взаимным индексом совпадений двух строк называют вероятность того, что символ, случайно выбранный из первой строки, совпадает со случайно выбранным символом второй строки.

Алгоритм нахождения ключа. Рассмотрим две строки таблицы. Сдвинем алфавит одной из строк на s символов и вычислим взаимный индекс совпадений полученных строк. Т.к. каждая из этих двух строк получена сдвигом алфавита открытого текста, то максимум взаимного индекса совпадений будет наблюдаться при нулевом конечном относительном сдвиге.

Поэтому применяется следующий алгоритм: вычисляется взаимный индекс совпадений для различных s, ищется значение s, при котором взаимный индекс совпадений максимален. Тогда начальный относительный сдвиг строк будет равен (m — размер алфавита). Вычисляются относительные сдвиги между всеми парами строк. Т.к. сдвиги строк таблицы соответствуют сдвигам букв ключа, то остаётся перебрать m возможных ключей и выбрать из них наиболее правдоподобный.

# **Ход лабораторной работы**

1. Возьмём в качестве примера 3 английских текста длиной 10000 слов. Я взял отрывки из романов “Лолита”, “Финансист” и “Я Эдичка” на английском языке. На сайте <https://www.blindtextgenerator.com/> сгенерируем 3 набора по 10000 случайных слов из разных словарей. Кроме того, напишем на языке C++ генератор случайной строки и сгенерируем ещё 2 текста из случайных букв.

2. Напишем на языке C++ компаратор (<https://github.com/Pavloffff/MAI_Crypto/blob/main/lab3/main.cpp>). Он будет выдавать процент совпадения букв в 2 текстах. При сравнении учитываются только латинские буквы, остальные символы не индексируются.

Тексты:

1. t1 - <https://github.com/Pavloffff/MAI_Crypto/blob/main/lab3/t1> (Я Эдичка)
2. t2 - <https://github.com/Pavloffff/MAI_Crypto/blob/main/lab3/t2> (Финансист)
3. t3 - <https://github.com/Pavloffff/MAI_Crypto/blob/main/lab3/t3> (Лолита)
4. t4 - <https://github.com/Pavloffff/MAI_Crypto/blob/main/lab3/t4> (dummy text 1)
5. t5 - <https://github.com/Pavloffff/MAI_Crypto/blob/main/lab3/t5> (random string 2)
6. t6 - <https://github.com/Pavloffff/MAI_Crypto/blob/main/lab3/t6> (random string 2)

3. В результате тестирования выяснилось, что:

1. Процент совпадений двух реальных текстов на английском языке составил приблизительно 6.3%, что соответствует реальным статистическим данным.
2. Сравнение осмысленного текста и текста из случайных букв выдало около 3.8%. Это лучше, чем при сравнении двух текстов из случайных букв, так как осмысленный текст не является случайным и некоторые буквы могут встречаться чаще.
3. Сравнения осмысленного текста и текста из случайных слов также выдали около 6.4%. Так как тексты “Я Эдичка” и “Лолита” слабо похожи друг на друга, то этот результат не отличается от первого.
4. При сравнении двух текстов из случайных букв был получен самый низкий результат - 3.8%. Это не удивительно, ведь вероятность подобного равна , что практически соответствует тестовым данным.
5. При сравнении двух текстов из случайных слов был получен результат 6.52%. Возможно, это связано с ограниченной длиной словарей.

4. На счет минимальной длины текста, достаточной для корректного сравнения двух текстов и выявления схожести их происхождения у меня есть следующие соображения. Чем больше длина текста, тем корректнее сравнения. Минимальной я бы назвал длину в 500-1000 слов, так как при такой длине частотность появления символа английского языка или случайность его появления будет весьма заметной.

5. Протокол тестирования:

lyutoev@ialyutoevlaptop:/mnt/f/crypto/lab3$ ./a.out

Enter 1st file name: text\_people\_1.txt

Enter 2nd file name: text\_people\_2.txt

6.36465

lyutoev@ialyutoevlaptop:/mnt/f/crypto/lab3$ ./a.out

Enter 1st file name: text\_people\_1.txt

Enter 2nd file name: text\_people\_3.txt

6.27043

lyutoev@ialyutoevlaptop:/mnt/f/crypto/lab3$ ./a.out

Enter 1st file name: text\_people\_2.txt

Enter 2nd file name: text\_people\_3.txt

6.27453

lyutoev@ialyutoevlaptop:/mnt/f/crypto/lab3$ ./a.out

Enter 1st file name: text\_people\_1.txt

Enter 2nd file name: text\_blob\_1.txt

6.3152

lyutoev@ialyutoevlaptop:/mnt/f/crypto/lab3$ ./a.out

Enter 1st file name: text\_people\_1.txt

Enter 2nd file name: text\_blob\_2.txt

6.2964

lyutoev@ialyutoevlaptop:/mnt/f/crypto/lab3$ ./a.out

Enter 1st file name: text\_people\_2.txt

Enter 2nd file name: text\_blob\_3.txt

6.37531

lyutoev@ialyutoevlaptop:/mnt/f/crypto/lab3$ ./a.out

Enter 1st file name: text\_people\_1.txt

Enter 2nd file name: text\_random\_1.txt

3.81308

lyutoev@ialyutoevlaptop:/mnt/f/crypto/lab3$ ./a.out

Enter 1st file name: text\_people\_2.txt

Enter 2nd file name: text\_random\_3.txt

3.94033

lyutoev@ialyutoevlaptop:/mnt/f/crypto/lab3$ ./a.out

Enter 1st file name: text\_people\_3.txt

Enter 2nd file name: text\_random\_3.txt

3.767

# **Выводы**

В процессе выполнения данной ЛР я узнал о различных индексах совпадений для реальных текстах и случайно сгенерированных строк. С помощью такой, казалось бы, скудной информации можно взломать относительно сложный шифр Виженера. Поэтому данный шифр, как и большинство шифров простой замены, нельзя назвать криптостойким. Лучше воспользоваться алгоритмом RSA.

# **Список используемой литературы**

1. <https://ru.wikipedia.org/>

2. <https://habr.com/ru/articles/103055/>