# 3. Шифр Виженера

3.1 алгоритм шифрования

Алгоритм шифрования осуществляет работу над комутативным кольцом вычетов по модулю n, где n- мощность алфавита используемого языка.

Введем обозначения:

Отождествим буквы алфавита с числами:

Алгоритм шифрования в веденных обозначениях выглядит следующим образом:

Алгоритм расшифрования можно путем выражения из уравнения шифрования символа открытого текста:

Расмотрим програмную реализацию шифрования и расшифрования.

Данные алгоритмы реализует функция:

void crypt(FILE \*source, char\* mode, char\* key);

FILE \*source - дескриптор файла, подлежащий шифровке или расшифровке;

char\* mode - модификатор работы функции;

char\* key - указатель на область памяти в которой хранится ключ;

//я в ожидание окончания работы

3.2 Криптоанализ шифра Вижинера

Криптоанализ шифра Виженера проходит в два этапа:

1. Поиск длины ключа
2. Поиск по длине ключа самого ключевого слова

Для определения длины ключа воспользуемся индексом совпадения.

**Индексом совпадения** называют вероятность того, что две выбранные случайно буквы текста совпадут.

Тогда вероятность совпадения двух случайно выбранных букв с будет равна .

Тогда индекс совпадений примет вид:

В случае, когда текст известен можно посчитать численное выражение для введенных величин.

Пусть символ алфавита встречается в тексте раз, при этом обозначим общее количество элементов текста за N, в таком случае:

Для русского и английских языков данный индекс посчитан и составляет для английского языка 0,0662, а для русского 0,529.

Вернемся к определению длины ключа. Посмотрим, как шифруется текст. Для это разобьем открытый текст M по следующему правилу:

В таком случае рассматривая преобразования символов в определенном получим, что для всех символов алфавита определено некоторое смещение равное постоянное для всех элементов перестановки . В таком случае можем сказать, что для шифрования символов лежащих в реализуется подстановка:

где – перестановка шифрованных символов полученных из .

При этом индекс совпадений для не изменится, так как в общей сумме останутся те же слагаемые, но в этом случае они будут относится к другим буквам определенным подстановкой , а в силу биективности каждое слагаемое будет присутствовать и в новой сумме.

В таком случае алгоритм поиска длины ключа будет сводится к разбиению зашифрованного текста на соответствующие перестановки и подсчитывания для них индекса совпадения. Если индекс совпадения равен с определенной точностью заранее определенным индексам для используемого языка, то используемую в данной итерации длину ключа можно считать предположительно верной. После достаточного анализа закрытого текста из всех выбранных длин выберем ту, у которой самый большой индекс совпадений и при этом она делит большинство выбранных значений.

Данный алгоритм реализуется функцией:

int find\_Key\_lenth(FILE\* file, char mod, char\*\* kod);

FILE \*file - дескриптор файла, подлежащий шифровке или расшифровке;

char\* mod - модификатор работы функции;

char\*\* kod - указатель на область памяти в которой будет хранится зашифрованный текст для последующего анализа;

// я не знаю, что писать

Следующий этап в криптоанализе шифра Виженера, это определение самого ключа. Для этого воспользуемся индексом взаимного совпадения.

**Взаимный индекс совпадения** называют вероятность того, что две выбранные случайно буквы из двух текстов совпадут.

вероятность совпадения повторно выбранной из другого текста буквы с

Пусть символ алфавита встречается в первом тексте раз, при этом обозначим общее количество элементов первого текста за N, а для второго текста возьмём аналогичные обозначения :

В таком случае взаимный индекс совпадения примет вид:

Разобьем С на по ранее указанному правилу, в таком случае текст разобьется на следующие перестановки:

Введем для каждого такой перестановки перестановку по следующему правилу:

- количество символов в перестановке .

Вернемся к взаимному индексу совпадения. Для осмысленных текстов написанных на английском и русском языках индекс посчитан и составляет 0,066 , 00000 соответственно. В случае если один из текстов представляет набор случайных (подразумевается неосмысленный набор букв) символов, то индекс будет принимать значения близкие к числу 0,04 или меньше. Посмотрим выражение взаимного индекса для нашего разбиения:

Рассмотрим взаимный индекс совпадения для

При этом вероятность совпадения i-ой буквы алфавита в соответствующих открытых текстах для равна , в таком случае взаимный индекс совпадения примет вид:

Произведем замену индекса суммирования и так как индекс берется как остаток от деления, в таком случае получим

В таком случае получим окончательную формулу для вычисления взаимного индекса совпадения

В таком случае, поиск ключевого слова сводится к перебору всевозможных d для двух таких перестановок и поиска среди них удовлетворяющей условиям:

1. приблизительно совпадает со значением взаимного индекса совпадения для осмысленных текстов написанных на данном языке;

Для реализации алгоритма поиска ключевого слова зафиксируем значение первого символа ключа, а для остальных символов найдем их сдвиг относительно первого. В таком случае пользователю остается из переложенных n фраз выбрать самую подходящую.

Данный алгоритм реализуется функцией:

void attack(FILE\* f);

FILE \*file - дескриптор файла, подлежащий шифровке или расшифровке;