Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

КАФЕДРА ИНФОРМАТИКИ

Лабораторная работа № 1

**Информационные сети. Основы безопасности. Информационные сети. Основы безопасности.**

Выполнил студент гр. 753503

Муха Д.С.

Проверил

Протько М.И.

Минск, 2020

# 1. Введение

Целью данной лабораторной было реализовать программные средства шифрования и дешифрования текстовых файлов при помощи Шифра Цезаря, (шифра сдвига, кода Цезаря) и шифра Виженера.

**ЗАДАНИЕ:**

1. Изучить теоретические сведения.
2. Создать программы, читающие данные из файла и шифрующие (дешифрующие) их с помощью с помощью Шифра Цезаря и шифра Виженера.

2. Блок-схемы алгоритмов

**1) Шифр Цезаря:**



**2) Шифр Виженера**



**3. Теоретические сведения**

**Шифр Цезаря**

Шифр Цезаря, также известный, как шифр сдвига, код Цезаря или сдвиг Цезаря – один из самых простых и наиболее широко известных методов шифрования.

Шифр Цезаря – это вид шифра подстановки, в котором каждый символ в открытом тексте заменяется символом находящимся на некотором постоянном числе позиций левее или правее него в алфавите. Например, в шифре со сдвигом 4 А была бы заменена на Г, Б станет Д, и так далее.

Шифр назван в честь римского императора Гая Юлия Цезаря, использовавшего его для секретной переписки со своими генералами.

Шаг шифрования, выполняемый шифром Цезаря, часто включается как часть более сложных схем, таких как шифр Виженера, и все ещё имеет современное приложение в системе ROT13. Как и все моноалфавитные шифры, шифр Цезаря легко взламывается и не имеет практически никакого применения на практике.



Рисунок 1. Шифр Цезаря

***Математическая модель***

Если сопоставить каждому символу алфавита его порядковый номер (нумеруя с 0), то шифрование и дешифрование можно выразить формулами модульной арифметики:

y=(x+k)\ \mod\ n

x=(y-k+n)\ \mod\ n,

где ~x – символ открытого текста, ~y – символ шифрованного текста, ~n – мощность алфавита, а ~k – ключ.

С точки зрения математики шифр Цезаря является частным случаем аффинного шифра.

***Пример***

Шифрование с использованием ключа k = 3. Буква «Е» «сдвигается» на три буквы вперёд и становится буквой «З». Твёрдый знак, перемещённый на три буквы вперёд, становится буквой «Э», и так далее:

Исходный алфавит:

***АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ***

Шифрованный:

***ГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯАБ***

Оригинальный текст:

*Съешь же ещё этих мягких французских булок, да выпей чаю.*

Шифрованный текст получается путём замены каждой буквы оригинального текста соответствующей буквой шифрованного алфавита:

*Фэзыя йз зьи ахлш пвёнлш чугрщцкфнлш дцосн, жг еютзм ъгб.*

**Шифр Виженера**

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 такой 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

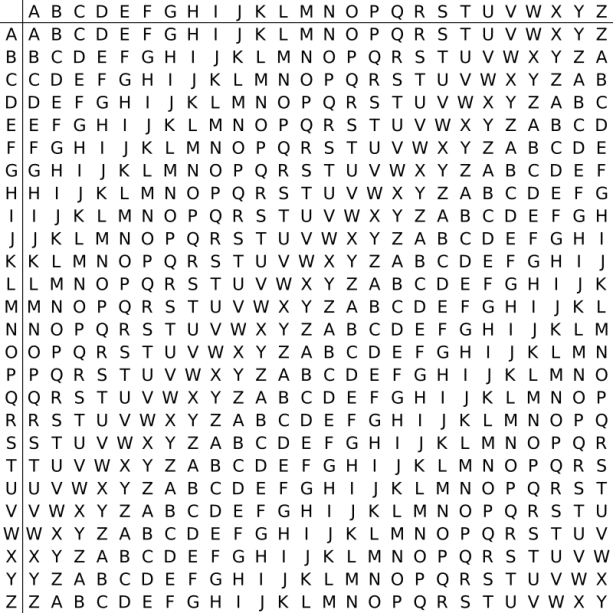


Рисунок 2. Квадрат Виженера

Расшифровывание производится следующим образом: находим в таблице Виженера строку, соответствующую первому символу ключевого слова; в данной строке находим первый символ зашифрованного текста. Столбец, в котором находится данный символ, соответствует первому символу исходного текста. Следующие символы зашифрованного текста расшифровываются подобным образом.

Если n — количество букв в алфавите, m j — буквы открытого текста, k j — буквы ключа, то шифрование Виженера можно записать следующим образом:

c j = m j + k j ( mod n )

И расшифровывание:

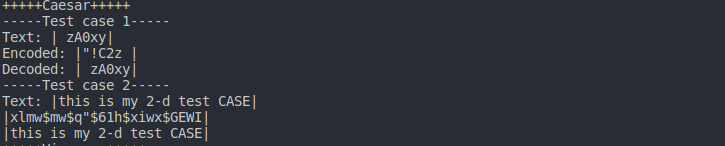
m j = c j − k j ( mod n )

В компьютере такая операция соответствует сложению кодов ASCII символов сообщения и ключа по некоторому модулю. Кажется, что если таблица будет более сложной, чем циклическое смещение строк, то шифр станет надежнее. Это действительно так, если ее менять чаще, например, от слова к слову. Но составление таких таблиц, представляющих собой латинские квадраты, где любая буква встречается в строке или столбце один раз, трудоемко и его стоит делать лишь на ЭВМ. Для ручного же многоалфавитного шифра полагаются лишь на длину и сложность ключа, используя приведенную таблицу, которую можно не держать в тайне, а это упрощает шифрование и расшифровывание.

# 4. Результаты выполнения программы

1. **Шифр Цезаря**

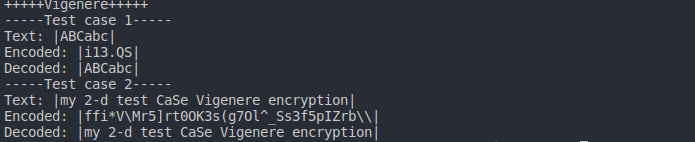
Шифратор/Дешифратор:



Дешифратор:

**2) Шифр Виженера:**

Шифратор/Дешифратор:



# 5. Выводы

В лабораторной работе номер один я восполнил знания о методах кодирования и декодирования Цезаря и Виженера, реализовал их программно.

# 6. КОД ПРОГРАММЫ

**1) Шифр Цезаря:**

package caesar

const (

LEFT\_BOUND = 32

RIGHT\_BOUND = 122

)

func Encode(text string, key int) string {

encoded := ""

for \_, v := range text {

if int(v) > RIGHT\_BOUND || int(v) < LEFT\_BOUND {

continue

}

if int(v)+key <= RIGHT\_BOUND {

encoded += string(int(v) + key)

} else {

tmp := RIGHT\_BOUND - int(v)

encoded += string(LEFT\_BOUND + (key - tmp - 1))

}

}

return encoded

}

func Decode(text string, key int) string {

decoded := ""

for \_, v := range text {

if int(v) > RIGHT\_BOUND || int(v) < LEFT\_BOUND {

continue

}

if int(v)-key >= LEFT\_BOUND {

decoded += string(int(v) - key)

} else {

tmp := int(v) - LEFT\_BOUND

decoded += string(RIGHT\_BOUND - (key - tmp - 1))

}

}

return decoded

}

**2) Шифр Виженера:**

package vigenere

import (

"unicode/utf8"

)

const (

LEFT\_BOUND = 32

RIGHT\_BOUND = 122

)

func Encode(text, key string) string {

encoded := ""

i := 0

module := (RIGHT\_BOUND - LEFT\_BOUND + 1)

for \_, v := range text {

if i > utf8.RuneCountInString(key)-1 {

i = 0

}

if int(v) < LEFT\_BOUND || int(v) > RIGHT\_BOUND {

encoded += string(v)

} else {

val := (((int(v) - LEFT\_BOUND) + (int(key[i]) - LEFT\_BOUND)) % module) + LEFT\_BOUND

encoded += string(val)

}

i++

}

return encoded

}

func Decode(text, key string) string {

decoded := ""

i := 0

module := (RIGHT\_BOUND - LEFT\_BOUND + 1)

for \_, v := range text {

if i > utf8.RuneCountInString(key)-1 {

i = 0

}

if int(v) < LEFT\_BOUND || int(v) > RIGHT\_BOUND {

decoded += string(v)

} else {

val := (int(v) - LEFT\_BOUND) - (int(key[i]) - LEFT\_BOUND)

if val < 0 {

val += module

}

decoded += string(val + LEFT\_BOUND)

}

i++

}

return decoded

}