# Практическая работа №4

# Потоковые шифры

**Цели:**

1. Рассмотреть симметричный алгоритм потокового шифрования RC4.

2. Приобрести навыки обращения к криптопровайдерам при разработке приложений.

3. Закрепить навыки использования криптографических классов платформы Microsoft .Net.

***Задание:*** Разработать приложение с пользовательским интерфейсом, которое позволит осуществлять шифрование и дешифрование открытых сообщений с помощью потокового алгоритма RC4, импортируемого из криптопровайдера ОС.

***Константы:*** отсутствуют.

***Переменные:*** KeyLen – Длина ключа(тип int)

## Исходные данные: Открытое сообщение «ПузановВладимир»

**Код MainForm:**

***Листинг 1 – Код MainForm:***

namespace RC4Application

{

public MainForm()

{

InitializeComponent();

// инициализация криптопровайдера

if (!CryptAcquireContext(ref hProv, null, null, 1, 0))

//если первая попытка неудачна, инициализируем криптопровайдер с созданием контейнера ключей

if (!CryptAcquireContext(ref hProv, null, null, 1, 8))

throw new CryptographicException

("Ошибка при инициализации криптопровайдера.");

// получение параметров длины ключа для алгоритма RC4

// создание объекта для приема значений параметров

PROV\_ENUMALGS\_EX data = new PROV\_ENUMALGS\_EX();

// задание длины буфера для приема параметров

int dlen = 100;

//получение параметров для первого криптоалгоритма, реализованного в криптопровайдере

if (!CryptGetProvParam(hProv, 22, ref data, ref dlen, 1))

throw new CryptographicException

("Ошибка при получении параметров длины ключа.");

// цикл до получения параметров криптоалгоритма RC4

while (data.szName != "RC4")

CryptGetProvParam(hProv, 22, ref data, ref dlen, 2);

// получение минимально возможной длины ключа

KeyLength = data.dwMinLen;

// создание массива данных для списка

ks = new int[12];

// индексы элемента списка и выделенного элемента

int i = 0, isel = 0;

// максимальная длина ключа и длина ключа по умолчанию

int ksmax = data.dwMaxLen, ksdef = data.dwDefaultLen;

// получение шага изменения возможной длины ключа

// задание длины буфера дли приема данных

dlen = 4;

//первое поле объекта data в этом случае содержит шаг изменения длины ключа

if (!CryptGetProvParam(hProv, 35, ref data, ref dlen, 0))

throw new CryptographicException

("Ошибка при получении шага изменения длины ключа.");

// заполнение списка возможных значений длины ключа

do

{

ks[i] = KeyLength;

// сохранение индекса для длины ключа по умолчанию

if (KeyLength == ksdef)

isel = i;

// увеличение возможной длины ключа

KeyLength += data.aiAlgid;

i++;

}

while (KeyLength <= ksmax);

// связывание списка с массивом данных

KeyLen.DataSource = ks;

// выделение элемента списка

KeyLen.SelectedIndex = isel;

Encrypt.Enabled = PlainText.Text.Length == 0;

Decrypt.Enabled = CipherText.Text.Length == 0;

}

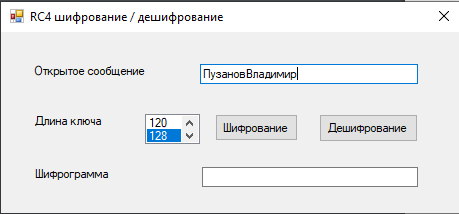


Рис.1 – Ввод открытого сообщения и выбор длины ключа

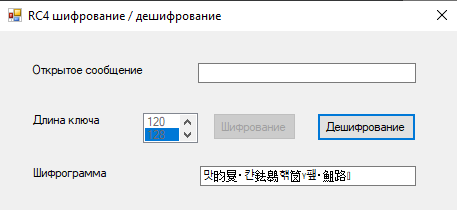


Рис.2 - Шифрование

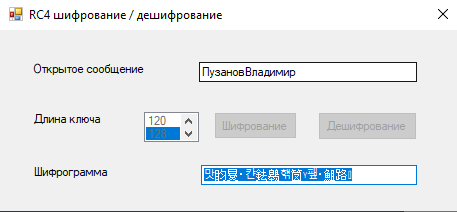


Рис.3 - Дешифрование

Выполнил студент Пузанов В. Е., ФИТУ 010304-КМСб-о22 Проверил ст. преподаватель каф. ПМ Черноиван Д.Н.