**Министерство Образования и Просвещения Республики Молдова**

**Бельцкий Государственный Университет им. ,,Алеко Руссо”**

**Индивидуальная работа**

**TEMA: Приложение для шифровки**

**Группа: IS11Z**

**Подготовил: Порческу Д.**

**Бельцы, 2020**

С момента создания первых шифров прошло не мало времени. Сегодня существуют множество алгоритмов, позволяющих сконструировать простые и эффективные шифры. Алгоритм шифрации, который выбрал я – это алгоритм RС4.

В этой индивидуальной работе показано, как шифровать и расшифровывать с помощью алгоритма RC4 (Rivest Cipher 4) в приложении C#.NET Windows Forms.

RC4 (от [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) Rivest cipher 4 или Ron’s code), также известен как ARC4 или ARCFOUR (alleged RC4) — [потоковый шифр](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%B2%D1%8B%D0%B9_%D1%88%D0%B8%D1%84%D1%80), широко применяющийся в различных системах защиты информации в компьютерных сетях

**Основные преимущества шифра:**

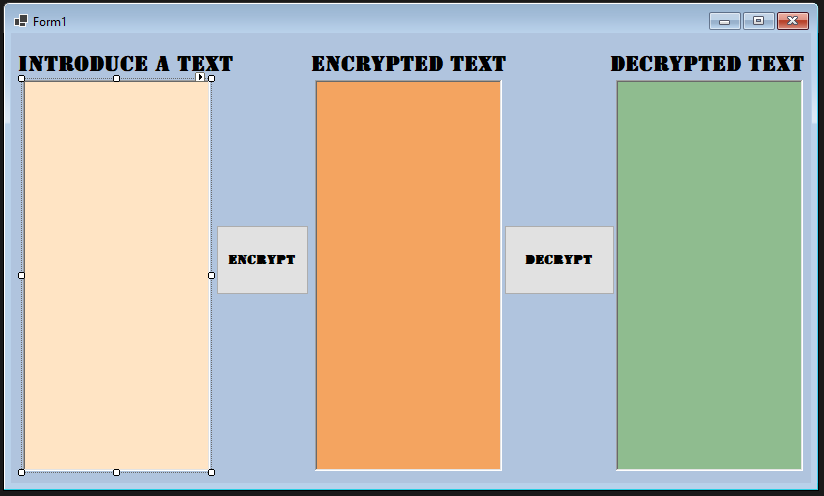
* высокая скорость работы;
* переменный размер ключа.

**RC4 довольно уязвим, если:**

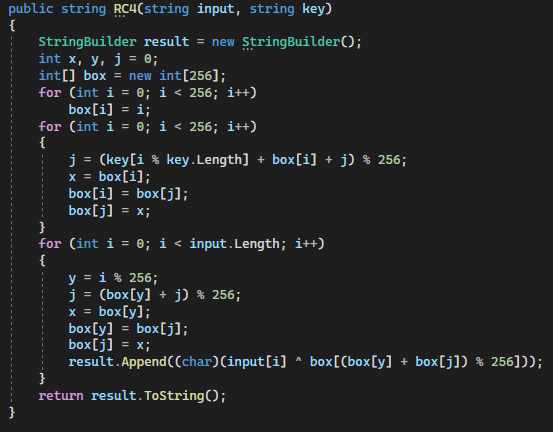
* используются не случайные или связанные ключи;
* один ключевой поток используется дважды.

Моя работа расскажет вам об алгоритме C# RC4, и продемонстрирует как шифровать и расшифровывать в C# с помощью него.

**1).** Для начала создаём приложение WinForms. И собираем визуальную часть. Несколько инпутов, кнопок и лэйблов:

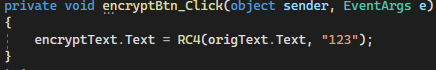


**2).** Далее пишем логику программы.

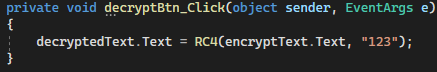


Это и есть наш алгоритм шифровки RC4. Но пока ещё мы не можем выводит шифрованное или зашифрованное сообщение. Поэтому добавляем ещё 2 блока кода, которые будут отвечать за шифровку и дешифровку.

**3).** Шифровка:

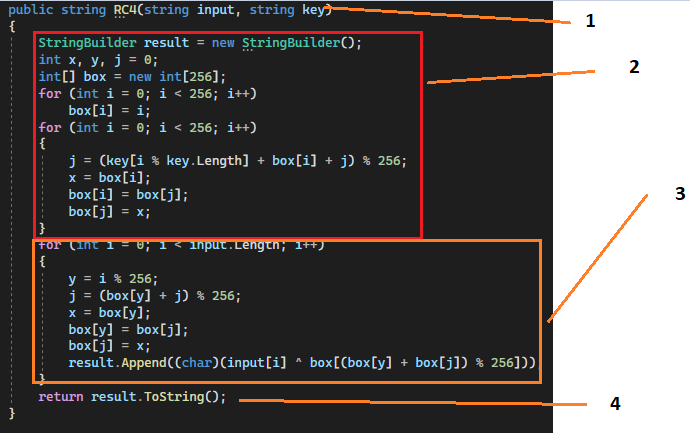


**4).** Дешифровка:

****

Оба события привязываем к соответствующим кнопкам – **decryptBtn** и **encryptBtn.**

Пара слов о том, как работает всё:



Алгоритм также известен как «**key-scheduling algorithm**» или «**KSA**». Этот алгоритм использует ключ, подаваемый на вход пользователем, сохранённый в **Key** (1), и имеющий длину L байт. Инициализация начинается с заполнения массива **box**, далее этот массив перемешивается путём перестановок, определяемых ключом. Так как только одно действие выполняется над **box**, то должно выполняться утверждение, что **S** всегда содержит один набор значений, который был дан при первоначальной инициализации (**box**[i] := i) (2).

В следующей части (3) алгоритма называется генератором псевдослучайной. Генератор ключевого потока RC4 переставляет значения, хранящиеся в **box**. В одном цикле RC4 определяется одно n-битное слово из ключевого потока. В дальнейшем ключевое слово будет сложено по модулю два с исходным текстом, которое пользователь хочет зашифровать, и получен зашифрованный текст.

В конце просто возвращаем результат (4).