МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

**Отчет по лабораторной работе №8**

**″Исследование потоковых шифров″**

Выполнила студентка 3 курса 5 группы Максимчикова Ю. С.

Проверила: Блинова Е. А.

Минск 2020

**Цель**: изучение и приобретение практических навыков разработки и использования приложений для реализации потоковых шифров.

**Задачи**:

* Закрепить теоретические знания по алгебраическому описанию, алгоритмам реализации операций зашифрования/расшифрования и оценке криптостойкости потоковых шифров.
* Разработать приложение для реализации указанных преподавателем методов генерации ключевой информации и ее использования для потокового зашифрования/расшифрования.
* Выполнить анализ криптостойкости потоковых шифров.
* Оценить скорость зашифрования/расшифрования реализованных шифров.
* Оценить скорость зашифрования/расшифрования реализованных способов шифров.
* Результаты выполнения лабораторной работы оформить в виде описания разработанного приложения, методики выполнения экспериментов с использованием приложения и результатов эксперимента.

**Теория.** Потоковый шифр (иногда говорят «поточный») – симметричный шифр, преобразующий каждый символ mi открытого текста в символ шифрованного, ci, зависящий от ключа и расположения символа в тексте. Термин «потоковый шифр» обычно используется в том случае, когда шифруемые символы открытого текста представляются одной буквой, битом или реже – байтом. Все потоковые шифры делятся на 2 класса: синхронные и асинхронные (или самосинхронизирующиеся). Основной задачей потоковых шифров является выработка некоторой последовательности (гаммы) для зашифрования, т.е. выходная гамма является ключевым потоком (ключом) для сообщения.

Синхронные потоковые шифры (СПШ) характеризуются тем поток ключей генерируется независимо от открытого текста и шифртекста. Главное свойство СПШ – нераспространение ошибок. Ошибки отсутствуют, пока работают синхронно шифровальное и дешифровальное устройства отправителя и получателя информации. Один из методов борьбы с рассинхронизацией – разбить отрытый текст на отрезки, начало и конец которых выделить вставкой контрольных меток (специальных маркеров).

Синхронные потоковые шифры уязвимы к атакам на основе изменения отдельных бит шифртекста. В самосинхронизирующихся потоковых шифрах символы ключевой гаммы зависят от исходного секретного ключа шифра и от конечного числа последних знаков зашифрованного текста. Основная идея заключается в том, что внутреннее состояние генератора потока ключей является функцией фиксированного числа предыдущих битов шифртекста. Поэтому генератор потока ключей на приемной стороне, приняв фиксированное число битов, автоматически синхронизируется с генератором гаммы. Недостаток этих потоковых шифров – распространение ошибок, так как искажение одного бита в процессе передачи шифртекста приведет к искажению нескольких битов гаммы и, соответственно, расшифрованного сообщения.

В данной лабораторной работе необходимо было разработать авторское приложение в соответствии с целью лабораторной работы. Для варианта номер 11 в качестве рассматриваемого метода был взят линейный конгруэнтный генератор с заданными параметрами для генерации ПСП и алгоритм RC4.

Реализация линейного конгруэнтного генератора представлена на рисунке 1.

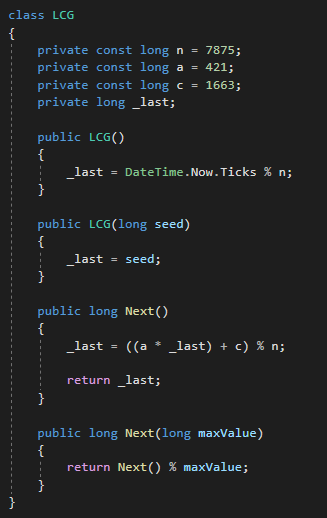


Рис. 1 – Линейный конгруэнтный генератор

Алгоритм RC4 разработан Р. Ривестом в 1987 г. Представляет собой потоковый шифр с переменным размером ключа. Здесь гамма не зависит от открытого текста.

Алгоритм RC4, как и любой потоковый шифр, строится на основе генератора псевдослучайных битов (генератора ПСП). На вход генератора записывается ключ, а на выходе читаются псевдослучайные биты. Длина ключа может составлять от 40 до 2048 бит.

Ядро алгоритма состоит из функции генерации ключевого потока. Другая часть алгоритма – функция инициализации, которая использует ключ переменной длины Ki для создания начального состояния генератора ключевого потока. В основе алгоритма – размер блока или слова, определяемый параметром n. Обычно n = 8, но можно использовать и другие значения. Внутренне состояние шифра определяется массивом слов (S-блоком) размером 2n. При n = 8 элементы блока представляют собой перестановку чисел от 0 до 255, а сама перестановка зависит от ключа переменной длины. Другими элементами внутреннего состояния являются 2 счетчика (каждый размером в одно слово; обозначим их i и j) с нулевыми начальными значениями. В основе вычислений лежит операция по mod 2n. Генератор ключевого потока RC4 переставляет значения, хранящиеся в S, и каждый раз выбирает различное значение из S в качестве результата. В одном цикле RC4 определяется одно n-битное слово K из ключевого потока, которое в последующем суммируется с исходным текстом для получения зашифрованного текста. Эта часть алгоритма называется генератором ПСП.

Байт K используется в операции XOR с открытым текстом для получения 8-битного шифртекста или для его расшифрования. Так же достаточно проста и инициализация S-блока. Этот алгоритм использует ключ, который подается на вход пользователем. Сначала S-блок заполняется линейно: S0 = 0, S1 = 1, …, S255 = 255. Затем заполняется секретным ключом другой 256-байтный массив. Если необходимо, ключ повторяется многократно, чтобы заполнить весь массив: K0, K1, …, K255. Далее массив S перемешивается путем перестановок, определяемых ключом.

Реализация функции шифрования алгоритма RC4 представлена на рисунке 2.

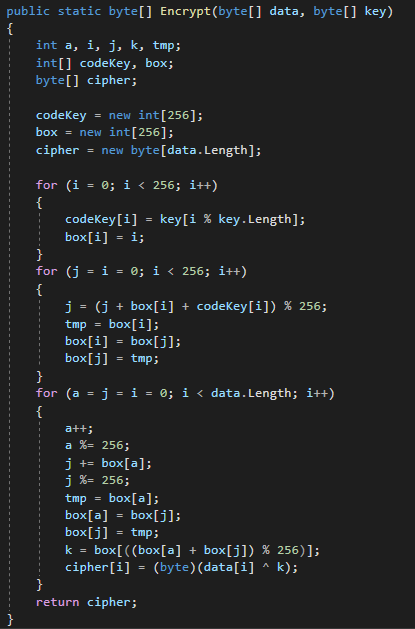


Рис. 2 – Функция шифрования алгоритмом RC4

Результат работы алгоритма и замеры времени выполнения операция шифрования и расшифрования сообщения Maximchikova Yuliya представлены на рисунке 3.

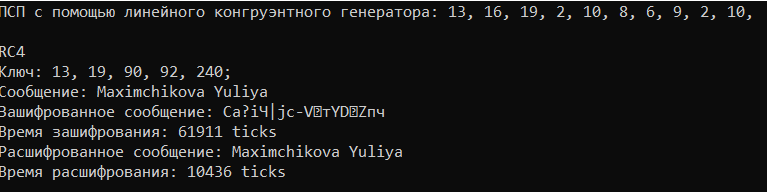


Рис. 3 – Результат работы приложения

Вывод: в результате данной лабораторной работы было разработано приложение для генерации ПСП с помощью линейного конгруэнтного генератора и для выполнения зашифрования и дешифрования текстов с помощью потокового шифра под названием RC4. Также был проведен анализ его криптостойкости и замеры времени работы операций шифрования и расшифрования с помощью этого алгоритма.