Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный технологический университет»

Факультет информационных технологий

Кафедра информационных систем и технологий

**Отчет по лабораторной работе №6**

**«Исследование потоковых шифров»**

Выполнил:

студентка 3 курса 2 группы

Черноок Ю. С.

Проверил:

ассистент

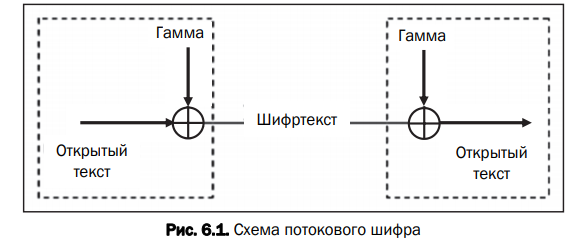
Копыток Д. В.

Минск 2021

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

**Теоретическая часть**

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). Необходимые начальные сведения об общих характеристиках и свойствах потоковых шифров можно найти в гл. 6 из [3], а также в [5]. 88 Лабораторная работа № 6 Основной задачей потоковых шифров является выработка некоторой последовательности (гаммы) для зашифрования, т. е. выходная гамма является ключевым потоком (ключом) для сообщения. В общем виде схема потокового шифра изображена на рис. 6.1.



Синхронные потоковые шифры (СПШ) характеризуются тем, что 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 битов 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.

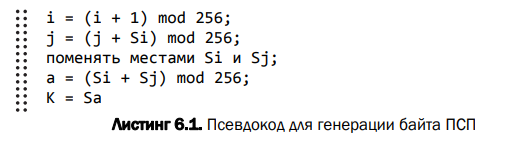
**Генератор псевдослучайных чисел на основе алгоритма BBS**

Широкое распространение получил алгоритм генерации ПСП, называемый алгоритмом BBS (от фамилий авторов: L. Blum, M. Blum, M. Shub) или генератором на основе квадратичных вычетов. Для целей криптографии этот метод предложен в 1986 г. Начальное значение x0 генератора вычисляется на основе соотношения x0 ≡ x2 mod n, где n, как и в генераторе на основе RSA, является произведением простых чисел p и q, однако в нашем случае эти простые числа должны быть сравнимы с числом 3 по модулю 4, т. е. при делении p и q на 4 должен получаться одинаковый остаток 3; число x должно быть взаимно простым с n; число n называют числом Блюма.

Выходом генератора на t-м шаге является младший бит числа xt:

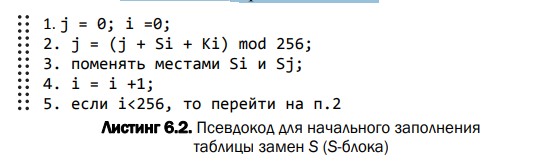
**Потоковый шифр RC4**

Алгоритм RC4 разработан Р. Ривестом в 1987 г. Он представляет собой потоковый шифр с переменным размером ключа. Здесь гамма не зависит от открытого текста [5]. 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 битов. 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. При n = 8 для генерации случайного байта выполняются операции, представленные листингом 6.1.



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. Действия выполняются в соответствии с псевдокодом, представленным листингом 6.2.



**Практическая часть**

Приложение 1 должно реализовывать генерацию ПСП в соответствии с вариантом.



Программная реализация алгоритма BBS представлена на рисунке 1.

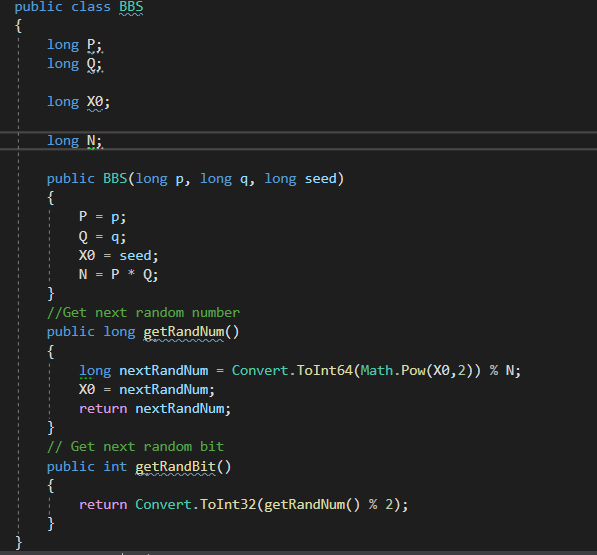


Рисунок 1 – Алгоритм BBS

Работа алгоритма BBS продемонстрирована на рисунке 2.

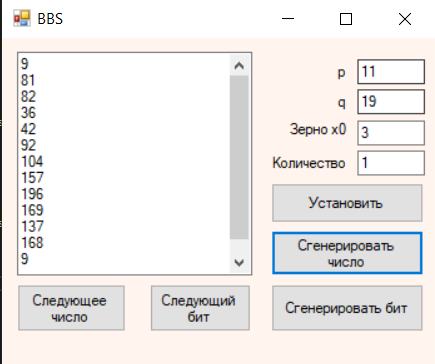
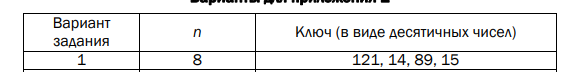


Рисунок 2 – Генерация ПСП на основе BBS

Приложение 2 должно реализовывать алгоритм RC4 в соответствии с вариантом.



Программная реализация алгоритма RC4 представлена на рисунке 3.

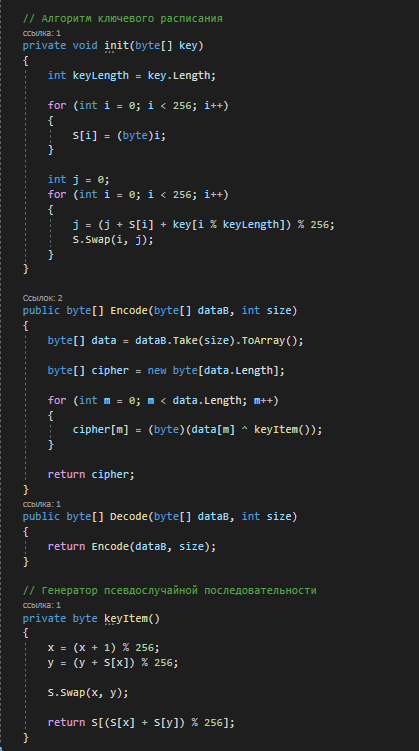


Рисунок 3 – Алгоритм RC4

Программный код обработчика нажатия на кнопки «Зашифровать» и «Расшифровать» представлен на рисунке 4.

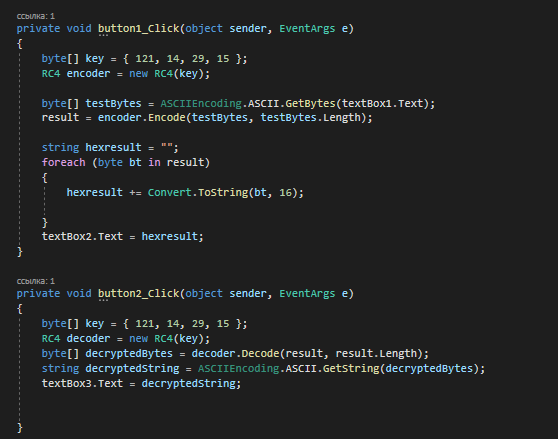


Рисунок 4 – Шифрование и расшифрование алгоритмом RC4

Работа алгоритма RC4 продемонстрирована на рисунке 5.

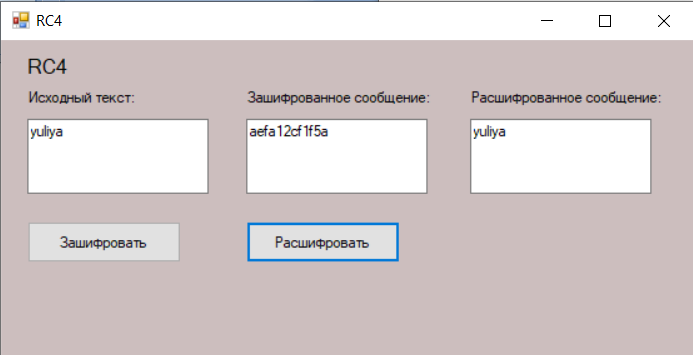


Рисунок 5 – Работа алгоритма RC4

Вывод: я изучила и приобрела практические навыки разработки и использования приложений для реализации потоковых шифров.