лУчреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий

Кафедра информационных систем и технологий

Специальность 1-40 01 01 Программное обеспечение информационных технологий

**Отчёт по лабораторной работе №8**

**«Исследование потоковых шифров»**

**Вариант 3**

Исполнитель:

Студент 3 курса группы 4

Гурина К. С.

Руководитель:

Ассистент Сазонова Д. В.

1. **Цель и задачи работы**

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

**Задачи:**

1. Закрепить теоретические знания по алгебраическому описанию, алгоритмам реализации операций зашифрования/расшифрования и оценке криптостойкости потоковых шифров.

2. Разработать приложение для реализации указанных преподавателем методов генерации ключевой информации и ее использования для потокового зашифрования/расшифрования.

3. Выполнить анализ криптостойкости потоковых шифров.

4. Оценить скорость зашифрования/расшифрования реализованных шифров.

5. Результаты выполнения лабораторной работы оформить в виде описания разработанного приложения, методики выполнения экспериментов с использованием приложения и результатов эксперимента

**2. Теоретические сведения**

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.

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, 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.

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.

Ключевые последовательности (случайные последовательности (СП), либо псевдослучайные последовательности (ПСП)) вырабатываются специальными блоками систем потокового шифрования – генераторами.

Часто используемый алгоритм генерирования (программно или аппаратно) ПСП реализуется на основе так называемого линейного конгруэнтного генератора, описываемого следующим рекуррентным соотношением:

xt+1 ≡ (a\*xx + c) mod n,

где xt и xt+1– соответственно t-й (предыдущий) и (t + 1)-й (текущий, вычисляемый) члены числовой последовательности; а, с и n – константы. Период такого генератора (период ПСП) не превышает n.

Алгоритм RC4 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 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 с нулевыми начальными значениями. В основе вычислений лежит операция по mod2n.

**3. Ход работы**

**Практическое задание:**

1. Разработать авторские многооконные приложения в соответствии с целью лабораторной работы. При этом можно воспользоваться готовыми библиотеками либо программными кодами, реализующими заданные алгоритмы.

Приложение 1 должно реализовывать генерацию ПСП в соответствии с вариантом из таблицы.

Приложение 2 должно реализовывать алгоритм RC4 в соответствии с вариантом из табл. 6.8, а также дополнительно выполнять оценку скорости выполнения операций генерации ПСП. В качестве шифруемого сообщения может быть выбран произвольный текст.

**Ход работы**

Начальное значение *x0* генератора вычисляется на основе соотношения

*,*

где n является произведением простых чисел p и q, однако в нашем случае эти простые числа должны быть сравнимы с числом 3 по модулю 4; число x должно быть взаимно простым с n; число n называют числом Блюма.

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

Для реализации алгоритма генерации псевдослучайных последовательностей с помощью алгоритма BBS была создана одноименная функция, представленная на рисунке 3.1.

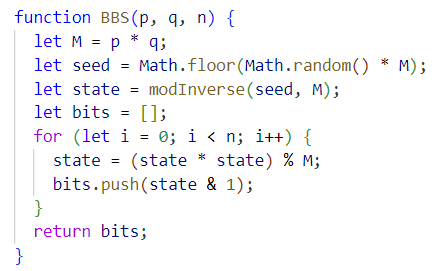


Рисунок 3.1 – Код генерации ПСП с помощью BBS

Для работы алгоритма необходимо выполнить следующие действия:

* инициализировать *S*-блок таблицы начальных замен;
* сгенерировать на основании *S*-блока псевдослучайные числа для ключей *K*;
* выполнить операцию сложения по модулю 2 каждого 8-битного ключа *K* с открытым текстом для получения шифротекста.

Для работы этой функции были реализованы дополнительные функции:

* Функция gcd вычисляет наибольший общий делитель (НОД) двух чисел a и b с использованием алгоритма Евклида.
* Функция modInverse вычисляет обратное числу **a** по модулю **m** с использованием расширенного алгоритма Евклида.
* Функция convertToDecimalGroups преобразует массив битов в строку десятичных чисел.

В соответствии с вариантом, на вход алгоритма подаются следующие числа: *p* и *q* - по согласованию с преподавателем, *x* = 256. Результат работы алгоритма, создающий 16-битное псевдослучайное число, представлен на рисунке 3.2.

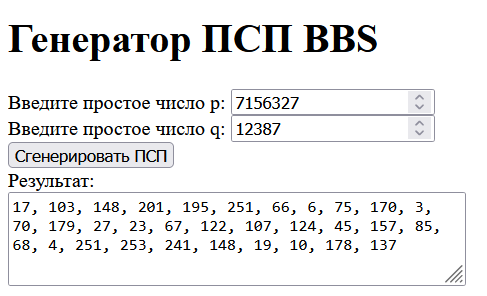


Рисунок 3.2 – Генерации ПСП с помощью BBS

Алгоритм RC4 является потоковым шифром, который используется для шифрования данных, обеспечивая высокую скорость и простоту реализации. Он включает два основных этапа: настройка ключа (KSA) и генерация псевдослучайного потока (PRGA). Ниже представлен общий алгоритм RC4.

Этап 1: Настройка ключа (KSA) представлен на рисунке 3.3.

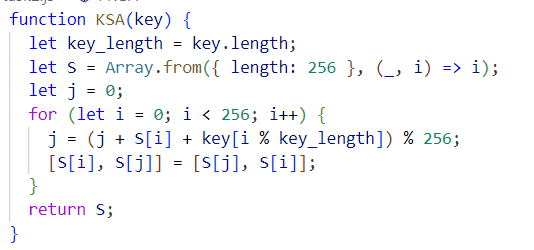
****

Рисунок 3.3 – Код функции KSA

Этап 2: Генерация псевдослучайного потока (PRGA)

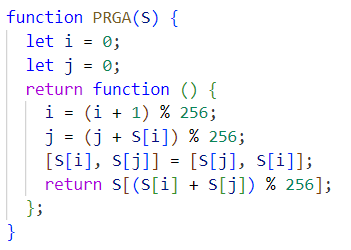


Рисунок 3.4 – Код функции PRGA

Функции для шифрования с помощью алгоритма PC4 представлены на рисунке 3.5.



Рисунок 3.5 – Код функций для шифрования с помощью алгоритма RC4

Результат работы функции можно увидеть на рисунке 3.6.

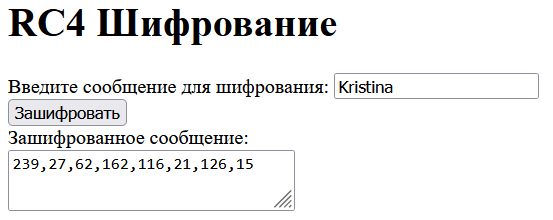


Рисунок 3.6 – Результат шифрования с помощью алгоритма RC4.

**Вывод**

В ходе выполнения лабораторной работы был изучен принцип реализации потоковых шифров. Также было разработано приложение для генерации ПСП на основе квадратичных вычетов, а также приложение, выполняющее шифрование и расшифрование с помощью шифра RC4.