Министерство образования Республики Беларусь

Учреждение образование

«Белорусский государственный технологический университет»

Кафедра информационных систем и технологий

**Отчет к лабораторной работе**:

«Исследование потоковых шифров»

Выполнил:

студент 3 курса 7 группы ФИТ

Салимон Павел Дмитриевич

Проверил:

Берников В. О.

Минск 2020

1. **Теоретические сведения**

**1.1 Классификация и общие свойства потоковых шифров**

*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 [2], а также в [4].

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.

**1.2 Генераторы ключевой информации**

Потоковый шифр максимально должен имитировать одноразовый блокнот. В соответствии с этим ключ должен по своим свойствам максимально походить на случайную числовую последовательность.

Ключевые последовательности (*случайные последовательности*, СП, либо *псевдослучайные последовательности*, ПСП) вырабатываются специальными блоками систем потокового шифрования – генераторами. В РБ в настоящее время действует стандарт СТБ 34.101.47-2017 "Информационные технологии и безопасность. Алгоритмы генерации псевдослучайных чисел" [31].

Стандарт устанавливает криптографические алгоритмы генерации псевдослучайных чисел. Алгоритмы стандарта могут применяться для построения ключей, синхропосылок, одноразовых паролей, других непредсказуемых или уникальных параметров криптографических алгоритмов и протоколов. Стандарт применяется при разработке, испытаниях и эксплуатации средств криптографической защиты информации.

**1.3 Линейный конгруэнтный генератор**

Часто используемый алгоритм генерирования (программно или аппаратно) ПСП реализуется на основе так называемого *линейного конгруэнтного генератора*

Если параметры *a*, *b* и *c* выбраны правильно, то генератор будет порождать случайные числа с максимальным периодом, равным *c*. При программной реализации значение *с* обычно устанавливается равным 2b-1 или 2b, где *b* – длина слова в битах.

Достоинством линейных конгруэнтных генераторов псевдослучайных чисел является их простота и высокая скорость получения псевдослучайных значений. Линейные конгруэнтные генераторы находят применение при решении задач моделирования и математической статистики, однако в криптографических целях их нельзя рекомендовать к использованию, так как специалисты по криптоанализу научились восстанавливать всю последовательность ПСЧ по нескольким ее значениям.

Генератор практически не используются в криптографии в силу низкой криптостойкости. Тем не менее, полезны для решения задач моделирования.

Комбинации нескольких (чаще двух) линейных конгруэнтных генераторов позволяют значительно повысить период ПСП. Б. Шнайер, например, приводит данные о том, как на 32-разрядных ПК реализовать генератор в виде комбинации двух, каждый из которых обеспечивает период соответственно 231 – 85 и 231 – 249, а комбинированный генератор позволяет достичь периода ПСП, равного произведению указанных чисел.

**1.4 *Генератор ПСП на основе регистров сдвига***

Достаточно распространенным является использование *регистров сдвига* (РС) в качестве генераторов ПСП в силу простоты реализации на основе цифровой логики. РС с линейной обратной связью (РСЛОС) состоит из двух частей: собственно РС и функции обратной связи. На рис. 6.2 представлена общая схема РС с линейной обратной связью. Функция обратной связи реализу-ется с помощью сумматоров сложения по модулю два

Само название конструкции Фейстеля (сеть) означает ее *ячеистую* топологию. Формально одна ячейка сети соответствует одному раунду зашифрования или расшифрования сообщения.

При зашифровании сообщение разбивается на блоки одинаковой (фиксированной) длины (как правило – 64 или 128 бит).

Полученные блоки называются *входными*. В случае, если длина входного блока меньше, чем выбранный размер, то блок удлиняется установленным способом.

Каждый входной блок шифруемого сообщения изначально делится на два подблока одинакового размера: левый (*L*0) и правый (*R*0). Далее в каждом *i*-ом раунде выполняются преобразования в соответствии с формальным представлением ячейки сети Фейстеля.

**1.5 Генератор псевдослучайных чисел на основе алгоритма *RSA***

Собственно алгоритм RSA разработан для систем асимметричного за-шифрования/расшифрования и будет более детально рассмотрен с практиче-ской точки зрения ниже.

Генератор же ПСП на основе RSA устроен следующим образом. Последо-вательность генерируется с использованием соотношения

Начальными параметрами служат *n*, большие простые числа *p* и *q* (причем *n* = *p*\**q*), целое число *е*, взаимно простое с произведением (р – 1)\*( *q* –1), а также некоторое случайное начальное значение, *x0.*

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

Безопасность генератора опирается на сложности взлома алгоритма RSA, т. е. на разложении числа *n* на простые сомножители.

**1.6 Генератор псевдослучайных чисел на основе алгоритма BBS**

Широкое распространение получил алгоритм генерации ПСП, называе-мый 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*

где *n*, как и в генераторе на основе RSA, является произведением простых чи-сел *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 одина-ковый остаток: 3; число *x* должно быть взаимно простым с *n*; число *n* называют *числом Блюма*.

**1.7 RC4**

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 состо-яние 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 внут-реннего состояния являются 2 счетчика (каждый размером в одно слово; обо-значим 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.

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: *S*0 = 0, *S*1 = 1, …, *S*255 = 255. Затем заполняется секретным ключом другой 256-байтный массив. Если необходимо, ключ повторяется многократно, чтобы заполнить весь массив: *K*0, *K*1, …, *K*255. Далее массив *S* перемешивается путем перестановок, определяемых ключом. Действия выполняются в соответствии с псевдокодом, представленным листингом.

1. **Практическая часть**

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. Общий интерфейс программы представлен на рисунке (Рисунок 2.1).

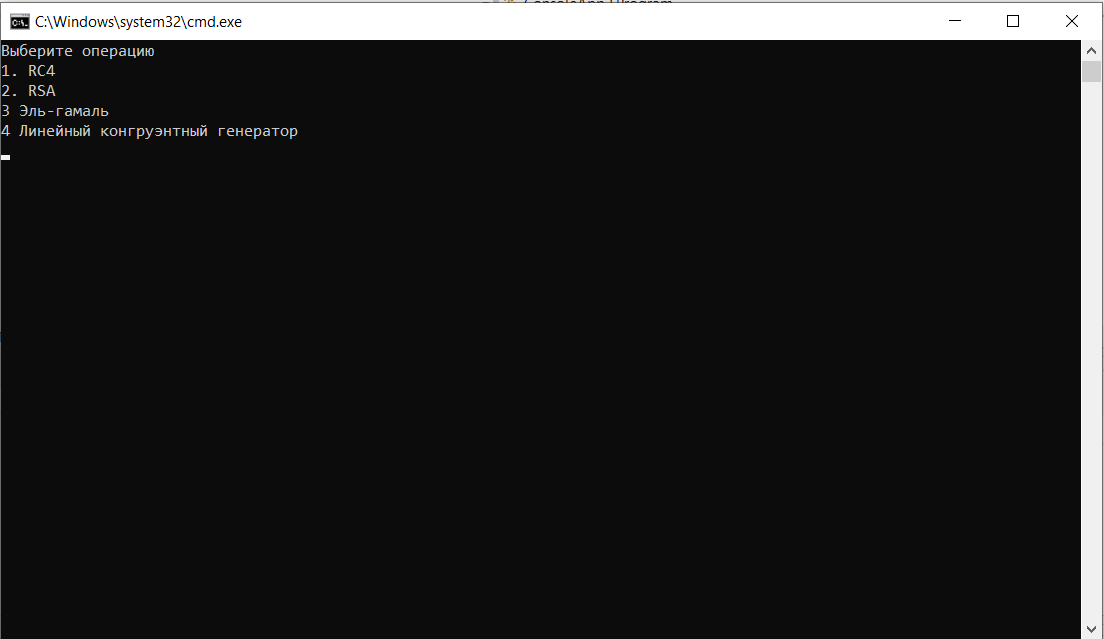


Рисунок 2.1 – Общий интерфейс программы

Реализация алгоритма RC4 приведена на рисунке (Рисунок 2.2)



Рисунок 2.2 – Алгоритм RC4

Генерация ПСП Линейным конгруэнтным генератором представлена на рисунке (Рисунок 2.3)

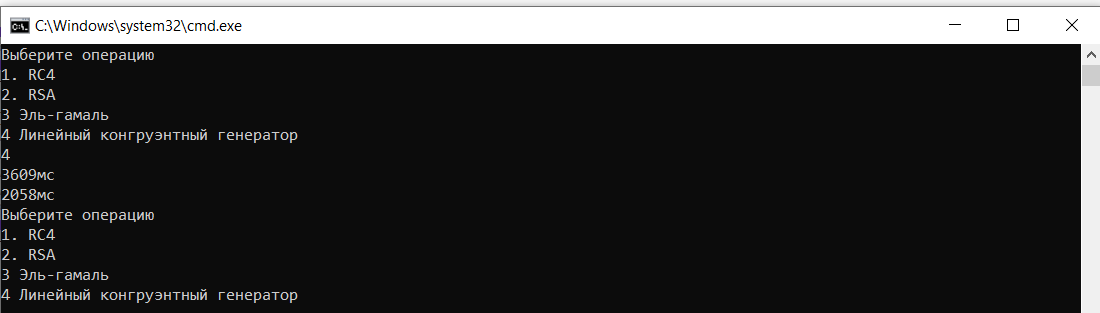


Рисунок 2.3 – Генератор ПСП

Вывод: В данной лабораторной работе были изучены теоретические сведения об поточных шифрах, а также разработано программное средство в соответствии с заданием, реализующее работу алгоритма RC4, а также реализовано программно генерация ПСП на основе линейного конгруэнтного генератора.