МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий

Кафедра информационных систем и технологий

Специальность Информационные системы и технологии

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ НА ТЕМУ:**

**Исследование потоковых шифров**

Ф.И.О.

Савельев Дмитрий Витальевич

Преподаватель

Берников Владислав Олегович

Минск 2021

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.

# 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).

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. 1.1.

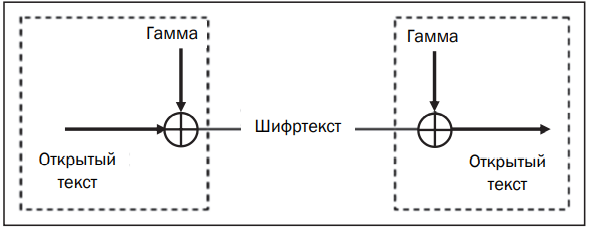


Рисунок 1.1 ­– Схема потокового шифра

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.

## 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 Беларуси в настоящее время действует стандарт СТБ 34.101.47–2017 «Информационные технологии и безопасность. Алгоритмы генерации псевдослучайных чисел».

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.

Указанный стандарт определяет базовые понятия в рассматриваемой предметной области:

• случайные числа (последовательности) – последовательность элементов, каждый из которых не может быть предсказан (вычислен) только на основе знания предшествующих ему элементов данной последовательности;

• псевдослучайные числа – последовательность элементов, полученная в результате выполнения некоторого алгоритма и используемая в конкретном случае вместо последовательности случайных чисел.

## 1.2.1. Линейный конгруэнтный генератор

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

|  |  |
| --- | --- |
|  | (1.1) |

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.

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 ПСП по нескольким ее значениям.

Генератор практически не используется в криптографии в силу низкой криптостойкости. Тем не менее он полезен 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.

## 1.2.2. 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. 1.2 представлена общая схема РС с линейной обратной связью. Функция обратной связи реализуется с помощью сумматоров сложения по модулю два (элементы XOR; на рис. 1.2 обозначены в виде кружочков со знаком сложения).

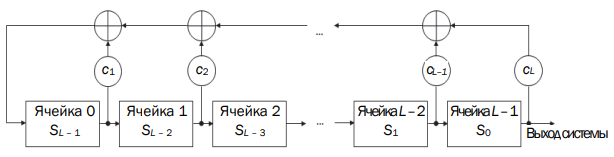


Рисунок 1.2 – Общая схема регистра сдвига с линейной обратной связью

РСЛОС строятся на основе примитивных порождающих полиномов (многочленов), которые мы подробно анализировали при изучении циклических помехоустойчивых кодов. Если многочлен является неприводимым, то период ПСП при ненулевом начальном условии (ненулевом состоянии) регистра будет максимально возможным: 2L – 1.

## 1.2.3. Генератор псевдослучайных чисел на основе алгоритма RSA

Алгоритм RSA разработан для систем асимметричного зашифрования/расшифрования и будет более детально рассмотрен с практической точки зрения ниже.

Генератор же ПСП на основе RSA устроен следующим образом. Последовательность генерируется с использованием соотношения

|  |  |
| --- | --- |
|  | (1.2) |

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

Выходом генератора на t-м шаге является младший бит числа xt. Безопасность генератора опирается на сложность взлома алгоритма RSA, т. е. на разложение числа n на простые сомножители.

## 1.2.4. 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

|  |  |
| --- | --- |
|  | (1.3) |

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:

|  |  |
| --- | --- |
|  | (1.4) |

С точки зрения безопасности важным является свойство рассмотренного генератора, заключающееся в том, что при известных p и q xt-й бит легко вычисляется без учета предыдущего (xt – 1) бита:

|  |  |
| --- | --- |
|  | (1.5) |

где а ≡ 2t mod ((p – 1)(q – 1)).

Алгоритм является сравнительно медленным. Для ускорения можно использовать не последний бит числа, а несколько последних битов. Однако понятно, что при этом алгоритм является менее криптостойким.

## 1.3. 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.

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.

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.

По утверждению, рассмотренный алгоритм устойчив к линейному и дифференциальному криптоанализу. Кроме того, при n = 8 этот алгоритм может находиться примерно в 21700 состояниях (256! · 2562).

При n = 16 алгоритм должен обладать большей в сравнении с n = 8 скоростью, но при этом начальные установки занимают гораздо больше времени – нужно заполнить 65 536-элементный массив.

Шифр RC4 применяется в некоторых широко распространенных стандартах и протоколах шифрования, таких как WEP, WPA и TLS, а также в Kerberos и др.

# 2. 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 с помощью линейного конгруэнтного генератора с a = 421, c = 1663, n = 7875.

Приложение 2 должно реализовывать алгоритм RC4 в соответствии с вариантом из таблицы, а также дополнительно выполнять оценку скорости выполнения операций генерации ПСП.

При запуске приложения запускается цикл из 20 итераций генерации псевдослучайного числа. А затем необходимо ввести фразу, которую мы хотим зашифровать с помощью потокового шифра RC4. Далее для каждого ключа из таблицы происходит шифрование и дешифрование с измерением быстродействия алгоритма.

Изображение выглядит как текст

Автоматически созданное описание

**Вывод.** Таким образом, в рамках данной лабораторной я изучил и разработал алгоритмы для генераторов ПСП, в частности линейный конгруэнтный генератор, потокового шифра RC4 и установил, что на примере шифров RC4 и DES, что блочные шифры заметно проигрывают потоковым в аспекте скорости шифрования и дешифрования.