Лабораторная работа 7

ИССЛЕДОВАНИЕ ГЕНЕРАТОРОВ ПСЕВДОСЛУЧАЙНЫХ ЧИСЕЛ

Цель работы: Знакомство с методами проектирования датчиков псевдослучайных чисел и генерации псевдобесконечных ключей.

1. Краткие теоретические положения

Случайные и псевдослучайные числа играют важную роль в криптографии. Например, при описании многих протоколов мы встречались с необходимостью генерировать большие случайные числа или слова, используемые как секретные ключи. Естественно, что задача генерирования последовательностей случайных чисел представляет большой интерес для разработчиков криптосистем. Более того, с развитием криптографии выяснилось, что многие фундаментальные проблемы этой науки тесно связаны с генерированием и тестированием случайных чисел.

Для получения случайных чисел можно использовать различные физические процессы, обладающие высокой производительностью и сравнительно легко сочетающиеся с компьютерными системами. Например, в качестве физических процессов, содержащих «случайную» составляющую, можно использовать шумы, возникающие в электрических цепях и их элементах, счетчики физических частиц, движения манипулятора-мыши в руке человека, работающего за компьютером, и т.д. Однако самый популярный способ получения случайных чисел не связан с наблюдениями за каким-либо сложным физическим процессом, а базируется на проведении вычислений. Получаемые таким способом числа называются псевдослучайными, что подчеркивает некоторое присущее им радикальное отличие от «истинно» случайных чисел.

К качеству случайных чисел, используемых в криптографии, предъявляются высокие требования. Прежде всего, требуется исключить возможные статистические отклонения от эталона: вероятности порождения различных значений должны быть в точности равны и генерируемые числа должны быть независимы. Для выявления таких отклонений используются специальные статистические методы, или тесты, которых к настоящему времени разработано довольно много.

Генераторы псевдослучайных чисел имеют максимальный период **p** до того, как последовательность начнет повторяться.

Одним из известных генераторов является ***линейный конгруэнтный генератор*** ПСЧ. Этот генератор вырабатывает последовательность псевдослучайных чисел $T_1, T_2, T_3, ..., T_m, ...$, используя соотношение

$$T_{i+1} = (a \times T_i + c) \bmod m,$$

где a и c — константы; T_0 — исходная величина, выбранная в качестве порождающего числа.

Указанное уравнение генерирует псевдослучайные числа с определенным периодом повторения, зависящим от выбранных значений a и c. Значение m обычно устанавливается равным 2^{b-1} или 2^b, где b — длина слова ЭВМ в битах. Значение a должно быть таким, что $a \pmod 4 = 1$ и a должно быть незначительно больше, чем $2^{b/2}$. Эти ограничения для генератора ПСЧ необходимы для генерации случайных чисел,

кажущихся случайными за счет поддержания большого периода.

Известны и другие схемы получения псевдослучайных чисел.

Метод Фибоначчи с запаздываниями (Lagged Fibonacci Generator) — один из методов генерации псевдослучайных чисел. Он позволяет получить более высокое «качество» псевдослучайных чисел.

Известны разные схемы использования метода Фибоначчи с запаздыванием. Один из широко распространённых фибоначчиевых датчиков основан на следующей рекуррентной формуле:

$$k_i = \begin{Bmatrix} k_{i-a} - k_{i-b}, & если\, k_{i-a} \geq k_{i-b} \\ k_{i-a} - k_{i-b} + 1, & если\, k_{i-a} < k_{i-b} \end{Bmatrix},$$

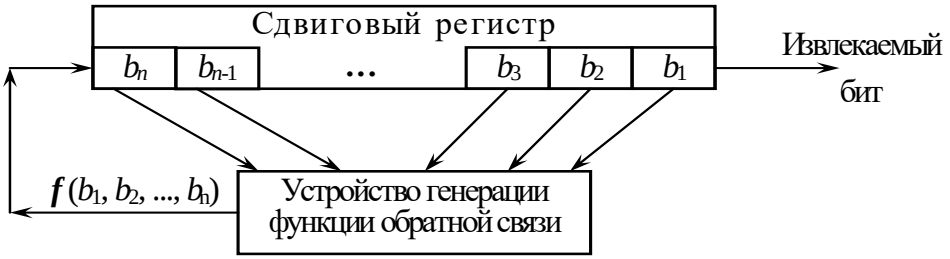
где k_i — вещественные числа из диапазона [0,1], a, b — целые положительные числа, параметры генератора. Для работы фибоначчиеву датчику требуется знать $\max\{a,b\}$ предыдущих сгенерированных случайных чисел. При программной реализации для хранения сгенерированных случайных чисел необходим некоторый объем памяти, зависящих от параметров a и b. Пример вычислений можно посмотреть, например, в [1].

Широкое распространение получил алгоритм генерации псевдослучайных чисел, называемый ***алгоритмом BBS*** (от фамилий авторов — L. Blum, M. Blum, M. Shub) или ***генератором с квадратичным остатком***. Для целей криптографии этот метод предложен в 1986 году. Он заключается в следующем. Вначале выбираются два больших простых числа p и q. Числа p и q должны быть оба *сравнимы* с 3 по модулю 4, то есть при делении p и q на 4 должен получаться одинаковый остаток 3. Далее вычисляется число $M = p \cdot q$, называемое целым число Блюма. Затем выбирается другое случайное целое число x, взаимно простое (то есть не имеющее общих делителей, кроме единицы) с M. Вычисляем $x_0 = x^2 \bmod M$. x_0 называется стартовым числом генератора.

На каждом n-м шаге работы генератора вычисляется $x_{n+1} = x_n^2 \bmod M$. Результатом n-го шага является один (обычно младший) бит числа x_{n+1}. Иногда в качестве результата принимают бит чётности, то есть количество единиц в двоичном представлении элемента. Если количество единиц в записи числа четное – бит четности принимается равным 0, нечетное – бит четности принимается равным 1. Пример вычислений можно посмотреть в [1].

Генераторы псевдослучайных чисел на основе сдвиговых регистров с обратной связью. В теории кодирования и криптографии широко применяются так называемые *сдвиговые регистры с обратной связью*. Они использовались в аппаратуре шифрования еще до начала массового использования ЭВМ и современных высокоскоростных программных шифраторов.

Сдвиговые регистры с обратной связью могут применяться для получения потока псевдослучайных бит. Сдвиговый регистр с обратной связью состоит из двух частей: собственно n-битного сдвигового регистра и устройства обратной связи (см. рис.).



Извлекать биты из сдвигового регистра можно только по одному. Если необходимо извлечь следующий бит, все биты регистра сдвигаются вправо на 1 разряд. При этом на вход регистра слева поступает новый бит, который формируется устройством обратной связи и зависит от всех остальных битов сдвигового регистра. За счет этого биты регистра изменяются по определенному закону, который и определяет схему получения ПСЧ. Понятно, что через некоторое количество тактов работы регистра последовательность битов начнет повторяться. Длина получаемой последовательности до начала ее повторения называется *периодом* сдвигового регистра.

Поточные шифры с использованием сдвиговых регистров достаточно долго использовались на практике. Это связано с тем, что они очень хорошо реализуются с помощью цифровой аппаратуры.

Простейшим видом сдвигового регистра с обратной связью является *линейный сдвиговый регистр с обратной связью* (linear feedback shift register – LFSR). Обратная связь в этом устройстве реализуется просто как сумма по модулю 2 всех (или некоторых) битов регистра. Биты, которые участвуют в обратной связи, образуют отводную последовательность. Линейные сдвиговые регистры с обратной связью или их модификации часто применяется в криптографии. Подробное описание и пример вычислений можно посмотреть в [1] или [2].

Алгоритм RC4. Алгоритм RC4 (описание алгоритма и пример вычислений см. в [1]) разработан Р.Ривестом специально как генератор потока ключевой информации с ключом переменной длины. Генераторы псевдослучайных чисел, построенные с помощью таких алгоритмов, как RC4, как правило, значительно быстрее генераторов, основанных на блочных шифрах. Алгоритм RC4 широко применяется в различных системах защиты информации, в компьютерных сетях. RC4 — фактически класс алгоритмов, определяемых размером его блока или слова – параметром n. Обычно n = 8, но можно использовать и другие значения.

2. Задание на работу

Изучить теоретические положения, конспект лекций, а также рекомендуемую литературу по данной теме. Получить вариант задания у преподавателя.

По своему варианту:

1) Реализовать программно генератор псевдослучайных чисел, имеющий в качестве выхода последовательность бит.

2) Разработать программу шифрования произвольных данных, записанных в файле, с помощью генерируемой последовательности бит, используемой в качестве гаммы.

3) Исследовать равномерность датчика ПСЧ (проверить гипотезу о равномерности распределения совокупности чисел, генерируемых датчиком ПСЧ).

4) Определить период датчика ПСЧ для заданных параметров.

3. Контрольные вопросы

1. Какие числа называют «псевдослучайными»?
2. Какими свойствами должен обладать генератор псевдослучайных чисел для использования в криптографических целях?
3. Какие генераторы псевдослучайных чисел Вы можете назвать?
4. Каким образом могут использоваться для получения псевдослучайных чисел сдвиговые регистры с обратной связью? Объясните их принцип работы.
5. В чем разница между генераторами случайных и псевдослучайных чисел?
6. Можно ли использовать генератор настоящих случайных чисел для получения гаммы при потоковом шифровании? Почему?
7. Для каких криптографических целей могут быть использованы генераторы настоящих случайных чисел?

4. Варианты для реализации ГПСЧ

Вариант 1. Двусторонний генератор «стоп-пошел». Описание см. в [2].

Вариант 2. ГПСЧ на основе сдвигового регистра с обратной связью (количество разрядов регистра: 12, $f = b_{11} \oplus b_5 \oplus b_2$).

Вариант 3. BBS (p = 25, q = 17)

Вариант 4. Генератор Геффа. Описание см. в [2].

Вариант 5. Генератор «стоп-пошел» (Stop-and-Go) Both-Piper. Описание см. в [2].

Вариант 6. Чередующийся генератор «стоп-пошел». Описание см. в [2].

Вариант 7. RC4 (n=12).

Вариант 8. Пороговый генератор с тремя LFSR. Описание см. в [2].

Вариант 9. Метод Фибоначчи с запаздыванием (a = 17, b = 5)

Вариант 10. RC4 (n=8).

Вариант 11. ГПСЧ на основе сдвигового регистра с обратной связью (количество разрядов регистра: 8, $f = b_8 \oplus b_4 \oplus b_3 \oplus b_2$).

Вариант 12. BBS (p = 23, q = 19)

Вариант 13. ГПСЧ на основе сдвигового регистра с обратной связью (количество разрядов регистра: 12, $f = b_{12} \oplus b_6 \oplus b_4 \oplus b_1$).

Вариант 14. Метод Фибоначчи с запаздыванием (a = 23, b = 7)

Вариант 15. RC4 (n=16).

Вариант 16. ГПСЧ на основе сдвигового регистра с обратной связью (количество разрядов регистра: 8, $f = b_8 \oplus b_6 \oplus b_5 \oplus b_4$).

Вариант 17. Самопрореживающийся генератора Рюппела. Описание см. в [2].

Вариант 18 Алгоритм Fish. Описание и параметры см. в [2].

Вариант 19 Алгоритм Pike. Описание и параметры см. в [2].

Вариант 20. Алгоритм Mush. Описание и параметры см. в [2].

Вариант 21. Прореживаемый генератор . Описание и параметры см. в [2].

Вариант 22. Самопрореживаемый (self-shrinking) генератор. Описание и параметры см. в [2].

Список рекомендуемой литературы

1. Басалова Г.В. Основы криптографии: учеб. пособие. Тула: Изд-во ТулГУ, 2009. 194 с.

2. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. М.: ТРИУМФ, 2002.