Министерство общего и профессионального образования РФ

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
"САМАРСКИЙ ГОСУДАРСТВЕННЫЙ АЭРОКОСМИЧЕСКИЙ
УНИВЕРСИТЕТ имени академика С.П.КОРОЛЕВА
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)"

СТАТИСТИЧЕСКИЙ АНАЛИЗ ПСЕВДОСЛУ-ЧАЙНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

Методические указания к
ЛАБОРАТОРНОЙ РАБОТЕ N 2

САМАРА 2023

Составители: к.т.н., доц. В.Н. Копенков

УДК 681.3

Генрерация и проверка случайных последовательностей:
Лабораторная работа N 2 / Самарский университет;
Самара, 2023. 14с.

В лабораторной работе изучается метод генерации случайных значений и проверки сгенерированных последовательностей на случайность на основе тестов NIST.

Лабораторная работа предназначена для студентов по курсу “Основы информационной безопасности” и для специалистов, проходящих курсы повышения квалификации.

Печатается по решению кафедры “Геоинформатика и информационная безопасность” Самарского государственного аэрокосмического университета имени академика С.П.Королева

Содержание:

Цель работы – обучение общим принципам статистического анализа на примере генераторов случайных чисел и их применение в ИБ.

Введение

Генерация высококачественных случайных чисел играет очень важную роль в самых разных приложениях – в криптографии, в области численного моделирования, других областях. В последние семьдесят лет, в связи с расширением области применения компьютеров и быстрым развитием электронных сетей связи, число таких приложений неуклонно растет. Как следствие, необходимо обладать высококачественными генераторами случайных чисел.

Основные области применения случайных чисел.

1) Криптография (Шифрование) – это математическая наука, практическая часть которой состоит из алгоритмов и протоколов, которые могут быть использованы для обеспечения конфиденциальности, подтверждения и проверки подлинности и целостности передаваемой информации. Криптографические алгоритмы различны, и выбор конкретного алгоритма зависит от условий эксплуатации (возможность использования компьютерных ресурсов) и решаемой задачи. Некоторые из них труднее взломать, но для их работы требуются существенные вычислительные мощности и сложное управление ключами. Другие алгоритмы легче взломать, но они и менее требовательны к вычислительным ресурсам компьютера, следовательно, лучше подходят для ряда приложений. Однако общим является тот факт, что все они требуют генерации истинно случайных чисел для создания ключей, и длина этих ключей зависит от схемы

шифрования. Многие схемы безопасности оказались взломаны или были серьезно нарушены, поскольку их генераторы случайных чисел не были достаточно случайными. Примером может служить генератор *A5/2*, взломанный за первые два дня своей эксплуатации.

Для того, чтобы гарантировать конфиденциальность передаваемого сообщения, отправитель создает зашифрованный текст путем сочетания текста, который необходимо передать, с ключом с помощью алгоритма шифрования. Затем этот зашифрованный текст передается по незащищенному (открытому) каналу связи получателю, который использует алгоритм расшифровки и ключ для расшифровки (либо это будет тот же закрытый ключ, заранее переданный по закрытому каналу связи, либо открытый ключ для случая асимметричной криптографии), чтобы восстановить исходный текст. В идеале злоумышленник не может расшифровать зашифрованный текст без ключа. Таким образом, сила системы шифрования в конечном итоге зависит от силы ключа или, что эквивалентно, от сложности для перехватчика угадать его. Непредсказуемость ключа напрямую зависит от энтропии в полученных случайных последовательностях. Следовательно, необходимо использовать достаточно хороший источник энтропии для генерации ключа, что приводит к необходимости проверки качества генератора случайных чисел.

2) Аутентификация – проверка подлинности. Эта процедура необходима, например, для того, чтобы при подключении клиента к серверу система удостоверилась в том, что он является тем, за кого себя выдает, и на самом деле имеет право на доступ к данным. На практике она часто использует случайные числа, например, при реализации протоколов строгой односторонней или двусторонней аутентификации, протокола аутентификации на основе асимметричного алгоритма.

3) Генераторы помех, шума, шумотроны. Данные технические приборы используются для генерации белого шума (постоянная случайная непериодическая на всем протяжении времени работы последовательность) в системах постановки радиопомех и системах радиоэлектронной борьбы (РЭБ). Также используются для наведения помех на работающие диктофоны и другие устройства звукозаписи, для подавления мобильной связи и т.п. Чаще всего для создания шумотронов применяются аппаратные генераторы случайных чисел, т.е. такие генераторы, которые для генерации случайного выходного сигнала используют какой-то непрерывный физический инертный процесс (например, ЛБВ (Лампа Бегущей Волны) – генератор с запаздывающей обратной связью).

4) Моделирование и имитация сложных систем, работающих в условиях неопределённости. Для таких систем были разработаны методы, опирающиеся на случайные числа. Здесь важно правильно имитировать распределения вероятностей случайных событий. Методы такой имитации базируются на равномерно распределённых случайных числах, и используемые генераторы случайных чисел должны обеспечивать это распределение с высокой точностью.

5) Лотереи и азартные игры. Распределение случайных чисел должно быть равномерное, а не псевдо-равномерное. Числа должны быть независимыми и должна отсутствовать возможность увеличения вероятности выигрыша, на основе анализа и обнаружения неравномерности распределения, либо зависимости генератора от набора параметров или внешних факторов.

1. Теоретические основы лабораторной работы.

1.1. Случайные числа и случайные последовательности

Случайное число – это число, порожденное процессом, исход которого непредсказуем и который не может быть впоследствии воспроизведен. Это определение хорошо подходит для условия, что есть некоторый «черный ящик» – называемый, как правило, датчиком или генератором случайных чисел, – который и выполняет задачу генерации. Если есть только какое-то одно число, то невозможно проверить, является ли оно результатом работы генератора случайных чисел или нет. С целью изучения случайности на выходе такого генератора необходимо рассматривать последовательности таких чисел. Определить, является ли последовательность бесконечной длины случайной или нет, вполне возможно: последовательность является случайной, если количество содержащейся в ней информации – в смысле теории информации Шеннона – тоже бесконечно. Интересен тот факт, что бесконечная случайная последовательность содержит все возможные конечные последовательности. Если мы имеем конечную последовательность чисел, то формально невозможно проверить, является ли она случайной или нет. Возможно только установить, имеет ли она статистические свойства случайных последовательностей или нет, т.е. все числа в этой последовательности должны быть равновероятны, некоррелированные, имеют постоянные математическое ожидание и дисперсию и т.п.

Наиболее важным является тот факт, что элементы случайной последовательности не должны быть коррелированы. Знание одного из чисел последовательности не должно помогать в прогнозировании других.

Таким образом, имея последовательность случайных чисел, созданную генератором случайных чисел, нам надо определить, является ли она действительно случайной либо как сильно она аппроксимирует случайную случайность. Для этого разрабатываются специальные статистические тесты случайности (например, линейка тестов NIST или DieHard), основанные на вычислении определенных статистических величин и сравнении их со средними значениями, которые были бы получены в случае по-настоящему случайной последовательности. Эти средние значения получаются из расчетов по модели идеального генератора случайных чисел.

Генератор случайных чисел – это устройство, которое создает последовательности чисел, описанные выше. Чаще всего на выходе таких датчиков – битовые последовательности. Существуют два основных вида генераторов: генераторы, связанные с определёнными алгоритмами, реализуемыми в программном обеспечении и физические генераторы, называемые также аппаратными генераторами. С точки зрения программного обеспечения генераторы первого типа генерируют так называемые псевдослучайные числа. Такие генераторы называются генераторами псевдослучайных чисел (ГПСЧ). Каждый ГПСЧ *рано или поздно зацикливается* – начинает повторять одну и ту же последовательность чисел. Длина циклов ГПСЧ зависит от самого генератора и равна примерно $2^{\frac{n}{2}}$, где n – размер внутреннего состояния в битах (линейные конгруэнтные и LFSR-генераторы обладают максимальными циклами длинны 2^n).

Большинство простых ГПСЧ хотя и обладают большой скоростью, но страдают от многих серьезных недостатков:

- слишком короткий цикл повторения;
- последовательные значения не являются независимыми;
- некоторые биты «менее случайны», чем другие;
- неравномерное одномерное распределение;

– обратимость процесса.

1.2. Методы проверки NIST случайных двоичных последовательностей

Существует множество методов проверки последовательностей на случайность. Основные методы представлены в линейках тестов NIST, DIEHARD и тестов Кнута. В данной лабораторной работе будет кратко рассмотрено три теста из линейки тестов NIST.

В основе данных тестов лежат понятия нулевой и альтернативной гипотез. Нулевая гипотеза – это гипотеза, основанная на предположении об отсутствии корреляции между объектами выборки. Альтернативная гипотеза – гипотеза, прямо противоположная нулевой. В случае проверки случайных чисел нулевая гипотеза – это утверждение о том, что рассматриваемая двоичная последовательность является истинно случайной, альтернативная – что нет. Проверка этих утверждений сводится к проверке на эталон статистики от фактически собранных данных. Различие реальных показателей от математической модели эталона должно попадать в границы выбранного порога.

В каждом тесте вычисляется P-значение – вероятность того, что генератор производит значения, сравнимые с эталоном. Ниже представлен обзор линейки тестов NIST с подробными математическими описаниями алгоритма. Если P-значение стремится к 1, то говорят, что генератор стремиться к идеальному. Если P-значение стремиться к 0 – генератор полностью предсказуем.

Частотный побитовый тест.

Если проверяемая последовательность достаточно случайна, то величина ее P-значения достаточно близка к единице. Этот тест оценивает эту близость.

Пусть дана последовательность ε длиной N и пусть «1» – это 1, а «0» – это -1. Тогда:

$$S_N = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} x_i \text{, где } x_i$$

$$= \begin{cases} 1, \text{если элемент проверяемой последовательности равен "1"} \\ -1, \text{если элемент проверяемой последовательности равен "0"} \end{cases}$$

С помощью дополнительной функции ошибок вычисляем P-значение:

$$P_{value} = \boldsymbol{erfc}\left(\frac{S_N}{\sqrt{2}}\right)$$

где $\boldsymbol{erfc}(x)$ – это дополнительная функция ошибок (присутствует в стандартной библиотеке ***math.h*** языка C и ***cmat.h*** языка C++).

Тест на одинаковые подряд идущие биты.

Тест заключается в поиске всех последовательностей одинаковых битов. Потом количество и размеры этих последовательностей анализируют на соответствие истинно случайной эталонной последовательности. Основная задача этого теста – установить насколько часто происходит смена «1» на «0» и обратно.

Сначала вычисляем долю единиц в последовательности ε длинной N:

$$\zeta = \frac{1}{N} \sum_{i=1}^{N} \varepsilon_i$$

где ε_i – это элементы проверяемой последовательности.

Далее проверяется условие $\left|\zeta - \frac{1}{2}\right| < \frac{2}{\sqrt{N}}$. Если оно истинно, то тест продолжается, иначе – P-значение считается равным нулю.

Далее вычисляется число знакоперемен V_N и на его основе P-значение:

$$V_N = \sum_{i=1}^{N-1} r_i\,, \qquad r_i = \begin{cases} 0, \text{если } \varepsilon_i = \varepsilon_{i+1} \\ 1, \text{если } \varepsilon_i \neq \varepsilon_{i+1} \end{cases}$$

$$P_{value} = \boldsymbol{erfc}\left(\frac{|V_N - 2N\zeta(1-\zeta)|}{2\sqrt{2N}\zeta(1-\zeta)}\right)$$

Тест на самую длинную последовательность единиц в блоке.

Исходная последовательность ε длинной N разбивается на блоки длиной M. Внутри блока ищется самая длинная последовательность из единиц, происходит ее оценка, которая сравнивается с аналогичной оценкой для эталонной случайной последовательности. Для последовательности в $N = 128$ бит длина блока M равна 8.

Далее в каждом блоке ищется максимальная длина подпоследовательности подряд идущих единиц (так в блоке, например «01001100» максимальная длина подряд идущих единиц равна 2, а в блоке «00010101» равна 1).

Далее считается статистика по разным длинам v_i следующим способом (для $M = 8$):

v_1 = { кол-во блоков с макс. длиной ≤ 1 }

v_2 = { кол-во блоков с макс. длиной = 2 }

v_3 = { кол-во блоков с макс. длиной = 3 }

v_4 = { кол-во блоков с макс. длиной ≥ 4 }

Например, для последовательности «11001100 00010101 01101100 01001100 11100000 00000010 01001101 01010001 00010011 11010110 10000000 11010111 11001100 11100110 11011000 10110010» (для удобства сразу разделена на блоки) эти значения v_i будут следующие:

v_1 = { кол-во блоков с макс. длиной ≤ 1 } = 4

v_2 = { кол-во блоков с макс. длиной = 2 } = 9

v_3 = { кол-во блоков с макс. длиной = 3 } = 3

v_4 = { кол-во блоков с макс. длиной ≥ 4 } = 0

Затем вычисляется Хи-квадрат (для данной лабораторной работы для последовательности длиной в 128 бит с блоками длиной 8):

$$\chi^2 = \sum_{i=0}^{3} \frac{(v_i - 16\pi_i)^2}{16\pi_i}$$

Распределение Хи-квадрат – это распределение суммы квадратов k независимых случайных величин; критерий хи-квадрат – это любая статистическая проверка гипотезы, в которой выборочное распределение критерия имеет распределение хи-квадрат.

Теоретические вероятности π_i задаются константами:

$\pi_0 = 0.2148$

$\pi_1 = 0.3672$

$\pi_2 = 0.2305$

$\pi_3 = 0.1875$

Далее вычисляется P-значение:

$$P_{value} = \boldsymbol{igamc}\left(\frac{3}{2}, \frac{\chi^2}{2}\right)$$

где $\boldsymbol{igamc}(x, y)$ – это неполная гамма-функция.

Для всех тестов справедливо, что если P-значение ≥ 0.01, то последовательность признается случайной.

2. Задание на лабораторную работу.

1) Необходимо сгенерировать псевдослучайные последовательности с помощью стандартных ГПСЧ языков C/C++ и Java.

В случае отсутствия среды Java-разработки разрешено использовать любую онлайн Java-машину, например, представленную по ссылке: https://www.onlinegdb.com/online_java_compiler.

Длина последовательностей – 128 бит, последовательности – бинарные.

Вид генератора и его параметры выбираются самостоятельно.

2) Необходимо написать программу на любом языке программирования, реализующую представленные три теста линейки NIST и проверить с их помощью сгенерированные последовательности:

a) Частотный побитовый тест.
b) Тест на одинаковые подряд идущие биты.
c) Тест на самую длинную последовательность единиц в блоке.

Ввиду сложности реализации неполной гамма-функции из данного теста, можно воспользоваться онлайн калькулятором неполной гамма-функции, для полученных значений, например:
https://www.danielsoper.com/statcalc/calculator.aspx?id=34.

3) Сделать вывод по полученным результатам и написать отчет по выполненной работе.

.

3. Выполнение лабораторной работы.

3.1. Общий план выполнения работы.

1. Изучить методы генерации случайных значений.
2. Изучить методы анализа случайный последовательностей.
3. Написать программу генерации случайно последовательности.
4. Написать программу реализации тестов NIST анализа случайных последовательностей.
5. Составить отчет о выполненной работе.
6. Сдать отчет преподавателю, ответить на контрольные вопросы, получить зачет по работе.

3.2. Этапы выполнения.

1. Генерация случайных последовательностей.

2. Проведение 3 тестов линейки NIST на сгенерированной последовательности.

3.3. Содержание отчета.

1. Результат выполнения первой части задания:
   а) Сгенерированная случайная последовательность;
   б) Программный код генерации;
2. Результат выполнения второй части задания:
   а) Программные коды тестов;
   б) Рассчитанные результаты тестов и полученные значения;
3. Выводы по результатам тестирования