Математические структуры и моделирование
2009, вып. 20, с. 134–140

УДК 003.26

ИССЛЕДОВАНИЕ ГЕНЕРАТОРОВ ПСЕВДОСЛУЧАЙНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ, ПОСТРОЕННЫХ НА ОСНОВЕ ХЭШ-ФУНКЦИЙ

М.И. Атмашкин, С.В. Белим

В работе проведен статистический анализ псевдослучайных последовательностей, формируемых на основе алгоритмов хэширования SHA-1 и MD5. В качестве метода исследования выбраны графические тесты.

1. Введение

Генераторы псевдослучайных последовательностей (ПСП) являются неотъемлемыми элементами большого количества вычислительных систем. Генератор псевдослучайных последовательностей (ГПСП) — это алгоритм, генерирующий последовательность чисел, элементы которой почти независимы друг от друга и подчиняются заданному распределению. ГПСП, как и поточные шифры, состоят из внутреннего состояния (обычно размером от 16 бит до нескольких мегабайт), функции инициализации внутреннего состояния ключом или зерном (англ. *seed*), функции обновления внутреннего состояния и функции вывода. Если $\{\gamma_i\}$ — псевдослучайная последовательность, полученная при использовании «хорошего» ГПСП, то три следующие задачи должны быть вычислительно сложными:

1) определение $(i-1)$-го элемента γ_{i-1} последовательности на основе известного фрагмента γ_i, γ_{i+1}, γ_{i+2}, ..., γ_{i+b-1} конечной длины b;
2) определение $(i+1)$-го элемента γ_{i+1} последовательности на основе известного фрагмента γ_{i-b+1}, ..., γ_{i-2}, γ_{i-1}, γ_i конечной длины b;
3) определение ключевой информации по известному фрагменту последовательности конечной длины.

Хеширование (англ. *hashing*) — преобразование входного массива данных произвольной длины в выходную битовую строку фиксированной длины. Такие преобразования также называются хеш-функциями, а их результаты называют хешем, хеш-кодом или дайджестом сообщения (англ. *message digest*). В общем случае однозначного соответствия между исходными данными и хеш-кодом нет.

Омский государственный университет.
E-mail: belim@univer.omsk.su

Поэтому существует множество массивов данных, дающих одинаковые хеш-коды – так называемые коллизии. Вероятность возникновения коллизий играет немаловажную роль в оценке «качества» хеш-функций.

2. Построение генератора псевдослучайных последовательностей

Формальное описание исследуемого далее ГПСП может быть представлено следующим образом:

1. Внутреннее состояние и его функция обновления

Внутренним состоянием ГПСП, основанного на хеш-функции MD5, является 128-битное хеш-значение H_i и ключевая фраза K произвольной длины (возможно и нулевой). Хеш-значение H_i вычисляется на основе предыдущего хеш-значения H_{i-1} в конкатенации с ключевой фразой:

$$Hi = MD5(H_{i-1}||K).$$

Ключевая фраза задается вначале один раз, но возможна маловероятная ситуация, когда H_i совпадет с H_{i-1}. В этом случае, чтобы избежать «застревания» генератора, следует ввести новую ключевую фразу. Для SHA-1 ситуация выглядит аналогично, за исключением того, что хеш-значение имеет размер 160 бит.

2. Функция инициализации внутреннего состояния зерном

В идеале зерно должно с равной вероятностью принять любое из значений от 0 до $2^{m\check{}1}$, где m – битовая разрядность используемой хеш-функции (на практике достаточно, чтобы каждое значение с ненулевой вероятностью). В данной работе для построения ГПСП использовалась комбинация двух методов: «счетчик тактов процессора» и «взаимодействие между потоками». Время (в наносекундах), прошедшее с момента загрузки системы, может быть получено как значение счетчика тактов процессора в момент инициализации внутреннего состояния, деленное на тактовую частоту процессора и умноженное на 10^9. Младшие 8 разрядов этого времени дадут один достаточно равномерно распределенный байт. Если сгенерировать несколько таких независимых байт (16 — для MD5, 20 — для SHA-1), а затем соединить в одно число, то получим зерно, отвечающее заданным требованиям. Равномерность распределения полученных величин следует из равномерного распределения байтов и способа получения зерна, а независимость следует из Теоремы 6 [2, стр. 65]. Независимости программно можно добиться, например, с помощью взаимодействия трех несимметричных по затратам процессорного времени потоков операционной системы. Первый поток – это основной поток программы (Main thread), при генерации зерна запускающий два других потока («Hard» и «Light»), один из которых состоит из бесконечного цикла, а другой генерирует нужное число байт (MD5 — 16 байт; SHA-1 — 20 байт) на основе 8-ми младших разрядов времени в наносекундах. Независимость этих байт обеспечивается сложным механизмом переключения между потоками, зависящим от большого числа факторов, быстрым изменением счетчика времени, дополнительной передачей управления на каждом шаге

цикла генерации, а также некоторыми примитивами синхронизации, которые не дают генерирующему потоку надолго захватывать процессор.

3. Функция вывода

На выходе генератора будет один бит, находящийся на произвольной (заданной заранее) позиции, отсчитываемой справа в текущем хеш-значении. Например, для MD5 — от 0 до 127, а для SHA-1 — от 0 до 159. «Независимость» этих битов обеспечивается сложностью хеш-функций. Далее эти биты последовательно записываются в файл и анализируются. Если предположить, что они действительно независимы и равномерно распределены, то независимыми и равномерно распределенными (по той же Теореме 6 [2, стр. 65]) будут и блоки из этих битов (в программе тестировались блоки до 16 бит).

3. Тестирование ПСП на случайность

Для сравнения ПСП с истинно случайной последовательностью применяются различные статистические тесты. Случайность – вероятностное свойство, это означает, что свойства случайной последовательности могут быть охарактеризованы и описаны в терминах вероятности. Вероятный результат статистических тестов, применяемых к истинно случайной последовательности, известен априорно и может быть описан в вероятностных терминах. Существует бесконечное число возможных статистических тестов, оценивающих присутствие или отсутствие «образца», который при обнаружении указал бы, что последовательность неслучайна. Поскольку существует много тестов, оценивающих, является ли последовательность случайной или нет, никакой определенный конечный набор тестов не считается на сегодняшний день «законченным».

Статистический тест формулируется для проверки определенной нулевой гипотезы H0 о том, что проверяемая последовательность является случайной. С этой нулевой гипотезой связана альтернативная гипотеза о том, что последовательность неслучайна. Для каждого применяемого теста получают заключение о принятии или отклонении нулевой гипотезы, основываясь на сформированной исследуемым генератором последовательности.

Для каждого теста должна быть выбрана подходящая статистика случайности для принятия или отклонения нулевой гипотезы. Согласно предположению о случайности, такая статистика имеет распределение возможных значений. Теоретическое эталонное распределение этой статистики для нулевой гипотезы определяется математическими методами. Из этого эталонного распределения определяется критическое значение. Во время проведения теста вычисляется значение тестовой статистики. Это значение сравнивается с критическим значением. Если значение тестовой статистики превышает критическое значение, то нулевая гипотеза для случайности отклоняется. В противном случае нулевая гипотеза принимается.

В данной работе был проведен ряд графических тестов, описанных в [3]. Рассмотрим результаты этих графических тестов:

1. Гистограмма распределения элементов

В исследуемой последовательности подсчитывается частота встречаемости

каждого элемента, после чего строится график зависимости числа появлений элементов от их численного представления (ASCII-значение для байтов). Для того чтобы последовательность удовлетворяла свойствам случайности, необходимо, чтобы в ней присутствовали все возможные элементы рассматриваемой разрядности, при этом разброс частот появления символов стремился к нулю. В противном случае последовательность не является случайной. Результаты теста приведены на рисунке 1 и удовлетворяют свойствам случайности.

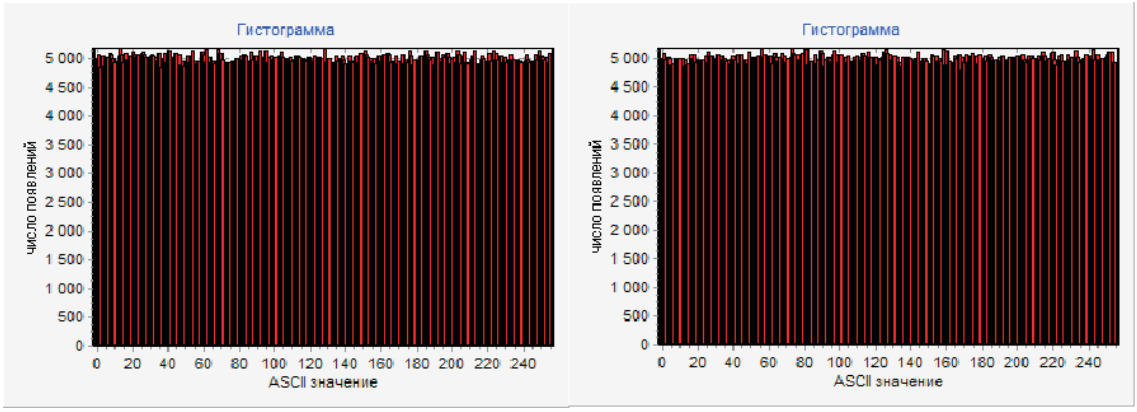

Рис. 1. Гистограмма распределения элементов. MD5 (слева) и SHA1 (справа)

2. Распределение на плоскости

На поле размером $(2^R-1)\times(2^R-1)$ (R – разрядность чисел исследуемой последовательности) наносятся точки с координатами $(\varepsilon_i;\varepsilon_i+1)$, где ε_i – элементы исследуемой последовательности ε, $i=1,\ldots,(n-1)$, n – длина последовательности. Если между элементами последовательности отсутствуют зависимости, то точки на поле расположены хаотично. Если на поле присутствуют зависимости, наблюдаются «узоры» — последовательность не является случайной. Результаты теста приведены на рисунке 2 и удовлетворяют свойствам случайности.

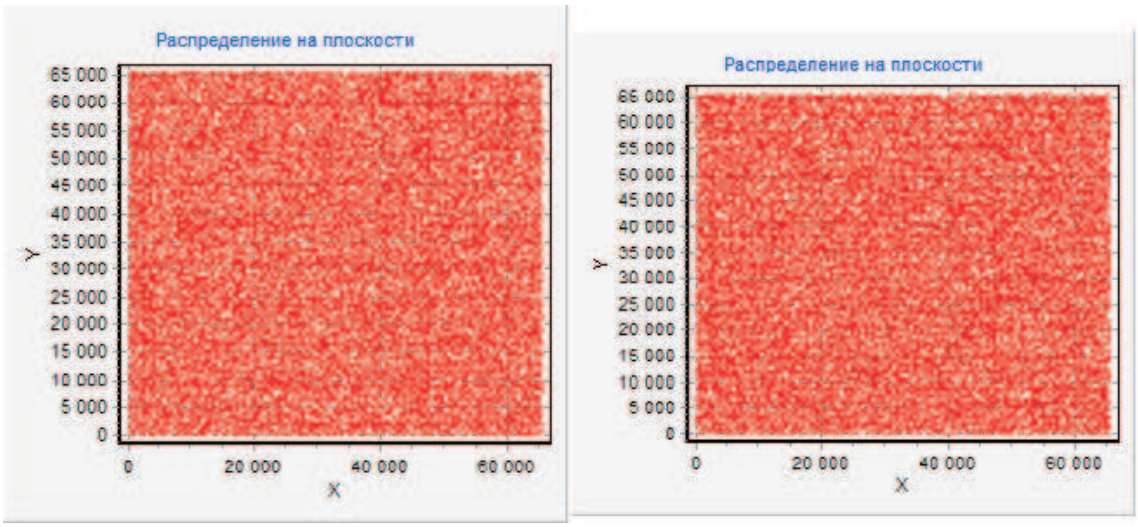

Рис. 2. Распределение на плоскости (R = 16 бит). MD5 (слева) и SHA1 (справа)

3. Проверка серий

Подсчитывается, сколько раз встречаются нули, единицы, серии-двойки (00, 01, 10, 11), серии-тройки (000, 001, 010, 011, 100, 101, 110, 111) и т. д. в битовом представлении исследуемой последовательности. Полученные результаты представляются в графическом виде. У последовательности, чьи статистические свойства близки к свойствам истинно случайной последовательности, разбросы между числом появлений серий каждого вида должны стремиться к нулю. В противном случае последовательность не является случайной. Результаты теста показаны на рисунке 3 и удовлетворяют свойствам случайности.

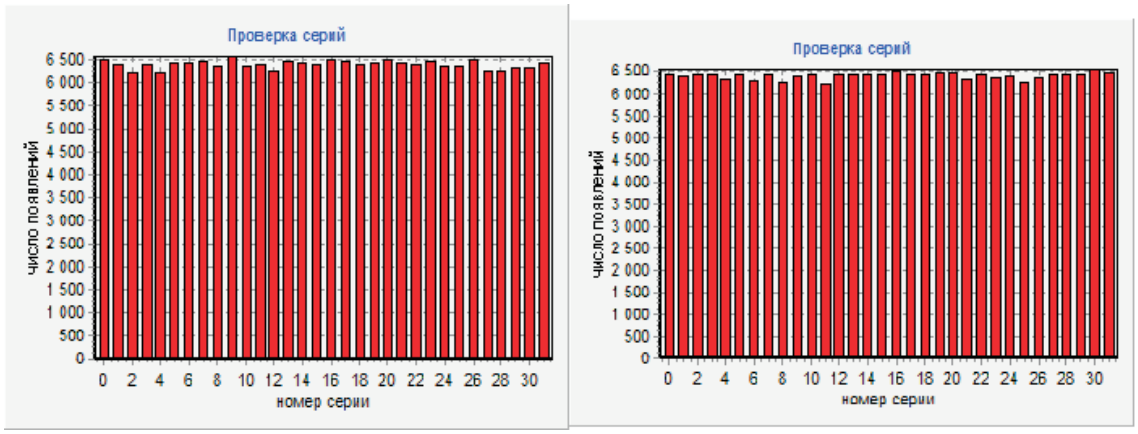

Рис. 3. Проверка серий длиной 5 бит. MD5 (слева) и SHA1 (справа)

4. Проверка на монотонность

Исследуемая последовательность графически представляется в виде следующих друг за другом непересекающихся участков невозрастания и неубывания элементов последовательности. У последовательности, чьи статистические

свойства близки к свойствам истинно случайной последовательности, вероятность появления участка невозрастания (неубывания) определенного размера зависит от его длины: чем больше длина, тем меньше вероятность. В противном случае последовательность не является случайной. Результаты теста показаны на рисунке 4 и удовлетворяют свойствам случайности.

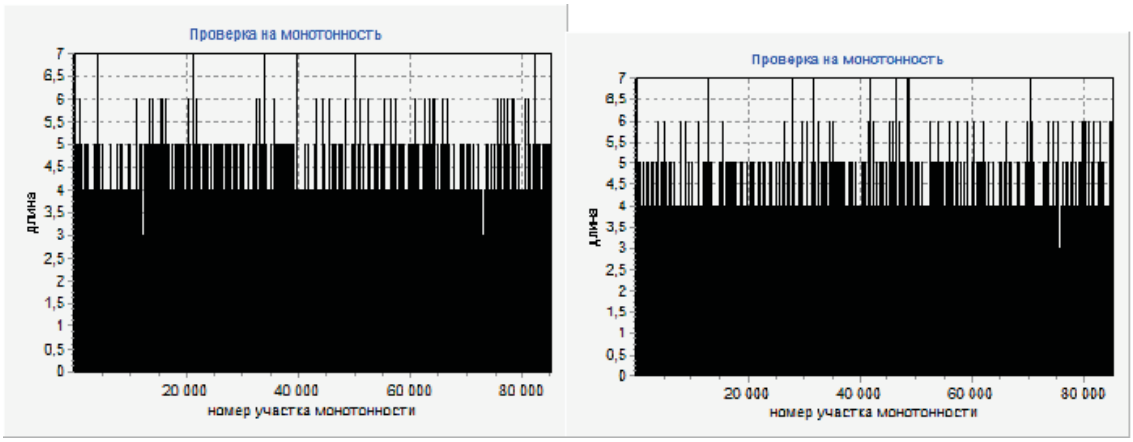

Рис. 4. Проверка на монотонность. MD5 (слева) и SHA1 (справа)

5. Профиль линейной сложности

Пусть $\varepsilon(n) = \varepsilon_1\varepsilon_2...\varepsilon_n$ — двоичная последовательность длины n. Последовательно рассматриваются подпоследовательности $\varepsilon(k)$, содержащие первые k элементов последовательности, и строится график зависимости линейной сложности L от длины подпоследовательности N. Линейная сложность вычисляется по алгоритму Берлекэмпа-Масси, который подробно описан в [1]. У последовательности, чьи свойства близки к свойствам истинно случайной последовательности, линия графика должна стремиться к линии $L = N/2$. В противном случае последовательность не является случайной. Результаты теста настоящего ГПСП показаны на рисунке 5 и удовлетворяют свойствам случайности.

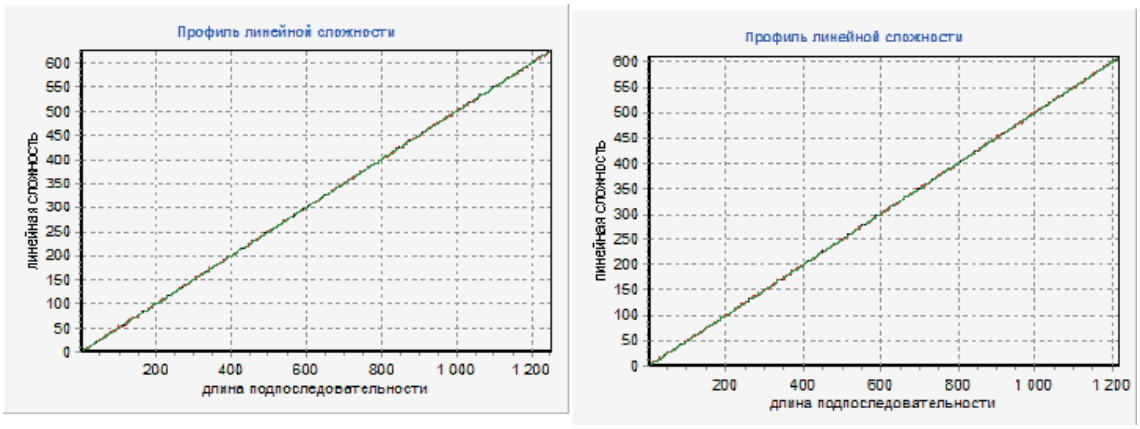

Рис. 5. Профиль линейной сложности. MD5 (слева) и SHA1 (справа)

6. Графический спектральный тест

Пусть $\varepsilon(n) = \varepsilon_1\varepsilon_2...\varepsilon_n$ — двоичная последовательность длины n. Преобразуем ее в последовательность $x = x_1x_2...x_n$, где $x_i = 2 \cdot \varepsilon_i \dot{-} 1$. Затем применим к x дискретное преобразование Фурье и получим последовательность гармоник:

$$S_j = \sum_{k=1}^{n} x_k e^{-i\frac{2\pi j}{n}(k-1)}.$$

Графически изобразим модули этих гармоник. У последовательности, чьи свойства близки к свойствам истинно случайной последовательности, число гармоник, длины которых значительно превышают среднюю длину гармоники, должно стремиться к 0. В противном случае последовательность не является случайной. Результаты теста ГПСП показаны на рисунке 6 и удовлетворяют свойствам случайности (в 3 раза среднюю длину превышают MD5 – 0.04%, SHA1 – 0.08%).

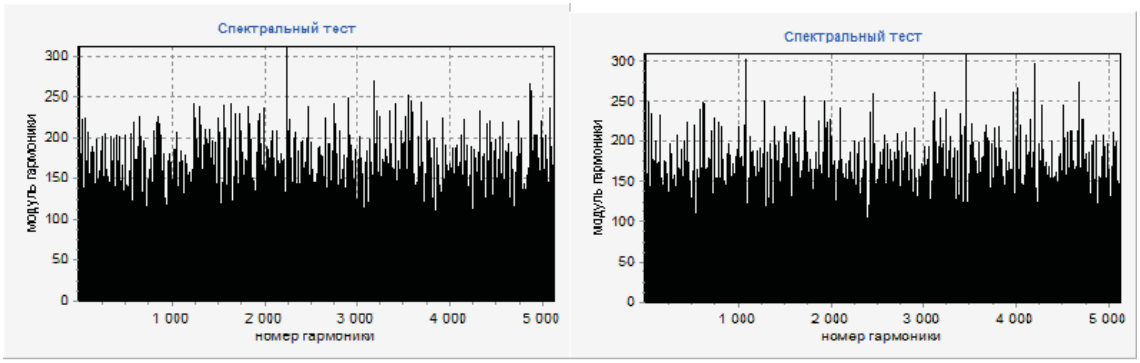

Рис. 6. Графический спектральный тест. MD5 (слева) и SHA1 (справа)

4. Заключение

Таким образом, генератор псевдослучайных чисел, построенный с использованием широко известных алгоритмов хэширования SHA-1 и MD5, является достаточно «хорошим» со статистической точки зрения. Данный факт весьма существенен при построении ряда систем защиты информации. Описанные хэш-функции достаточно легко поддаются аппаратной реализации, что позволяет строить качественные ГПСП на базе чипов с низкой тактовой частотой.

Литература

1. Берлекэмп, Э. Алгебраическая теория кодирования / Э. Берлекэмп. – М.: Мир, 1971. – 479 с.
2. Боровков, А.А. Теория вероятностей / А.А. Боровков. – М.: Наука, 1986. – 432 с.
3. Иванов, М.А. Теория, применение и оценка качества генераторов псевдослучайных последовательностей / М.А. Иванов, И.В. Чугунков. – М.: КУДИЦ-ОБРАЗ, 2003. – 240 с.