А.Ю. Григорьев, А.А Смагин

ОЦЕНКА СВОЙСТВ ГЕНЕРАТОРОВ СЛУЧАЙНЫХ ЧИСЕЛ НА ОСНОВЕ УЧЁТА СХОДСТВА ОДНОРОДНЫХ ЧАСТЕЙ ПОРОЖДАЕМОЙ ПОСЛЕДОВАТЕЛЬНОСТИ

(Ульяновский государственный университет)

Введение

В современном мире большое значение имеют генераторы случайных и псевдослучайных чисел (ГСЧ). Они широко применяются в различных задачах: моделирование. выборочные метод, численный анализ, программирование, криптография. Случайные и псевдослучайные последовательности бит, формируемые ГСЧ, имеют огромную роль для криптографии – от их качества зависит секретность информации. Поэтому задача создания хороших генераторов и эффективных методов их оценки представляет большой интерес.

Для оценки качества последовательностей бит используются различные статистические тесты. Для удобства исследований разработаны специальные пакеты статистических тестов, включающие алгоритмы тестирования и методы оценки результатов. Наиболее распространен и широко применяется пакет NIST STS[1], созданный Национальным институтом стандартов США, для оценки статистических свойств ГСЧ и алгоритмов шифрования.

Статистические тесты в NIST STS не являются абсолютно идеальными и постоянно находят более эффективные алгоритмы оценки, которые позволяют находить изъяны в ГСЧ. Так, в работе [2] был предложен новый статистический тест «Стопка книг», а в [3] показано, что он эффективнее тестов NIST STS. Поэтому постоянно существует необходимость создавать более эффективные алгоритмы оценки ГСЧ.

В данной работе предлагаются новые статистические тесты на основе вычисления расстояния Хэмминга и Левенштейна. Для оценки результатов используется метод, предложенный в тестах NIST STS.

Описание новых статистических тестов

Пусть некоторый источник порождает последовательность бит S длиной n бит ($S = \varepsilon_0\varepsilon_1 \dots \varepsilon_i \dots \varepsilon_{n-1}$). Последовательность разбивается на равные блоки длины m:

$S = S_1, S_2, \dots, S_n = \varepsilon_1\varepsilon_2 \dots \varepsilon_m, \varepsilon_{m+1}\varepsilon_{m+2} \dots \varepsilon_{2m}, \dots, \varepsilon_{n-m+1}\varepsilon_{n-m+2} \dots \varepsilon_n.$

В основе каждого метода лежит функция вычисления расстояния.

1) Функция вычисления расстояния Хэмминга имеет вид:

$$d(S_i, S_{i+1}) = \sum_{j=1}^{m} \varepsilon_{i\cdot m+j} \oplus \varepsilon_{i\cdot m+j+m}$$

2) Вычислить расстояние Левенштейна можно по следующей рекуррентной формуле:

$d(S_i, S_{i+1}) = D(M, N)$, где

$$D(M,N)=\begin{cases}0, & i=0, j=0\\ i, & j=0, i>0\\ j, & i=0, j>0\\ \min\{ & \\ D(i,j-1)+1, & \\ D(i-1,j)+1, & j>0\ , i>0\\ D(i-1,j-1)+m(S_1[i],S_2[j]) & \\ \} & \end{cases}$$

где *m(a,b)=0*, если *a=b*, в противном случае *m(a,b)=1*. *min{a,b,c}* возвращает наименьший из аргументов.

Для каждого предлагаемого теста используется два режима сравнения: пересекающиеся и непересекающиеся блоки бит. В первом случае каждый блок участвует в сравнении дважды (кроме первого и последнего) и сравниваются с шагом 1: S_1 и S_2, S_2 и S_3 и т.д. В втором случае каждый блок сравнивается только один раз: S_1 и S_2, S_3 и S_4 и т.д.

Функции расстояния Хэмминга и Левенштейна могут принимать $m+1$ значений в диапазоне от 0 до m. Распределение вероятности идеальной случайной последовательности бит для функции расстояния Хэмминга имеет биномиальное распределение и вычисляется по формуле:

$\begin{cases}H(i)=C_i^m, & i\geq 1\\ H(0)=1, & i=0\end{cases}$, где C_i^m – биномиальный коэффициент

Распределение вероятности для расстояния Левенштейна вычисляется экспериментальным способом.

В исследовании для тестов на основе расстояния Хэмминга и Левенштейна выбрана длина блока бит m равная 8 и 16 бит.

Описание процесса тестирования

Для проведения статистических тестов используется пакет NIST STS (The National Institute of Standards and Technology Statistical Test Suite), целью которого является определение меры случайности двоичных последовательностей, порождённых либо аппаратными, либо программными генераторами случайных чисел. Для предлагаемых статистических тестов на основе расстояния Хэмминга и Левенштейна разработаны средства оценки последовательностей бит в соответствии с методами, предложенными в NIST STS.

Процесс выполнения статистических тестов в пакете NIST STS выглядит следующим образом[4]:

1) Выбирается последовательность бит *S*. Её рекомендуемая длина составляет $100\cdot 10^6$ бит (согласно рекомендации программы NIST STS).

2) Для некоторых тестов задаются регулируемые параметры, которые зависят от длины последовательности бит (описание параметров приведено в документации к программе NIST STS).

3) Выполняются статистические тесты. Тестируемая последовательность S делится на m подпоследовательностей S_i ($S=\bigcup_{i=1}^{m} S_i$) длиной 10^6 бит каждая.

Порядок тестирования двоичной последовательности S_i (i=1..m) для каждого теста состоит из следующих шагов:

а) Выдвигается предположение о том, что данная двоичная последовательность S_i случайна;

б) По последовательности S_i вычисляется статистика теста $c(S_i)$;

в) С использованием специальной функции $f(x)$ и статистики теста вычисляется значение вероятности *p-value* = $f(c(S_i))$. В качестве $f(x)$ в зависимости от теста используются следующие функции:

$erfc = \frac{2}{\sqrt{\pi}} \int_{-x}^{\infty} e^{-u^2} du$ – дополнительная функция ошибок;

$\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{\frac{-u^2}{2}} du$ – стандартное нормальное распределение;

$igamc(a,x) = \frac{1}{\Gamma(a)} \int_{x}^{\infty} e^{-t} t^{a-1} dt,\ x \geq 0,\ a > 0$ – неполная гамма-функция, где $\Gamma(a) = \int_0^{\infty} t^{a-1} t\ dt$ – гамма-функция. Данная функция применялась для предложенных в работе статистических тестов.

г) Если *p-value* ≥ 0.01, то последовательность S_i считается случайной.

4) Завершение работы программы. Для каждого теста в отдельности создаётся отчёт, который содержит значения вероятности *p-value* и другие параметры, характерные для конкретного теста. Так же все суммарные расчетные данные размещаются в файле – отчёте, который отражает результаты всех пройденных тестов.

Для интерпретации результатов в NIST STS предложены два способа[1]:

1) Рассмотрение доли последовательностей, которые прошли статистические тесты. Последовательность S является случайной, если количество успешных тестов k последовательностей S_i удовлетворяет условию, описанному ниже:

$k \geq \left\lfloor \left(p' - 3\sqrt{\frac{\alpha\, p'}{m}} \right) m \right\rfloor$, где p' = 1- α, α = 0.01.

2) Распределение значений *p-value* для проверки на равномерность.

Множество значений *p-value* [0;1] разбивается на k интервалов с вероятностями, равными *1/k*. Дальше подсчитывается частоты F_i появления *p-value* для каждой категории, полученные в результате эксперимента. Вычисляется статистика:

$\chi^2 = \sum_{i=1}^{k} \frac{(F_i - m/k)^2}{m/k}$, где $m \geq 5k$

Далее вычисляется $p\text{-}value_T = igamc((k-1)/2, X^2/2)$.

Исследования

В оценке качества последовательностей бит принимали участие генераторы псевдослучайных последовательностей (линейный конгруэнтный генератор (LCG), квадратичный конгруэнтный генератор 1 (QCG-I), квадратичный конгруэнтный генератор 2 (QCG-II), кубический конгруэнтный генератор (CCG), генератор xor (XORG), modular Exponentiation Generator (MODEXPG), Secure Hash Generator (G-SHA1), Blum-Blum-Shub (BBSG), Micali-Schnorr Generator (MSG), функция rand() в stdlib.h) и архиваторы (7z, gz, rar, xz).

По результатам исследования в последовательностях, полученных из генераторов QCG-I, CCG, BBSG, MSG, G-SHA были выявлены небольшие статистические отклонения – 5-6 последовательностей из 100 не прошли тесты при норме 4 или меньше. В генераторах LCG, XORG и архиваторах gz, rar, xz тесты на основе вычисления расстояния Хемминга и Левенштейна выявили серьёзные статистические отклонения – более 10 последовательностей из 100 не прошли тесты. NIST STS также выявил отклонения в генераторах QCG-I, CCG, BBSG, MSG, G-SHA, LCG, XORG и архиваторах gz, rar, xz.

Можно сделать вывод, что тесты на основе вычисления расстояния Хэмминга и Левенштейна не хуже пакета NIST STS, так как забраковали те же самые генераторы. Так же стоит отметить, что предложенные в работе тесты имеют преимущества по скорости выполнения и ресурсным затратам, что позволяет использовать их для оценки ГСЧ в смартфонах, планшетах и других устройствах, не имеющих больших вычислительных мощностей.

Литература

1. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. NIST Special Publications 800-22. Revision 1.a April, 2010.

2. Б. Я. Рябко, А. И. Пестунов. «Стопка книг» как новый статистический тест для случайных чисел. Проблемы передачи информации. 2004. том 40, выпуск 1. С. 73–78

3. А. И. Миненко. Экспериментальное исследование эффективности тестов для проверки генераторов случайных чисел. Вестник СибГУТИ. 2010. № 4. С. 36-46

4. Чугунков И.В. Методы и средства оценки качества генераторов случайных последовательностей, ориентированных на решение задач защиты информации: Учебное пособие. М.: НИЯУ МИФИ, 2012. – 236 с.

Р.Р. Закиров

ЗАЩИТА ПРОГРАММ ОТ ОТЛАДКИ

(Казанский национальный исследовательский технический университет имени А.Н. Туполева – КАИ)

Процесс отладки позволяет злоумышленнику выяснить, по какому адресу выполняется программа, также подделать ее переменные в необходимые значения. Благодаря этому изучение программы конкурентами или пользователями, которые хотят воспользоваться платной программой бесплатно, облегчается в разы, чем при дизассемблировании и статичном изучении машинного кода программы.

Из этого следует, что защита от данного метода реверс-инжиниринга является наиболее актуальной для программистов. Поскольку сейчас популярна