Генератор псевдослучайной последовательности

Основная проблема классической криптографии долгое время заключалась в трудности генерирования непредсказуемых двоичных последовательностей большой длины с применением короткого случайного ключа. Существенный прогресс в разработке и анализе этих генераторов был достигнут лишь к началу шестидесятых годов.

Получаемые программно из ключа случайные или псевдослучайные ряды чисел называются на жаргоне отечественных криптографов *гаммой,* по названию — буквы у греческого алфавита, которой в математических записях обозначаются случайные величины.

Интересно отметить, что в книге «Незнакомцы на мосту», написанной адвокатом разведчика Абеля, приводится термин «гамма», который специалисты ЦРУ пометили комментарием «музыкальное упражнение?», т.е. в пятидесятые годы они не знали его смысла.

Следует при этом подчеркнуть, что заслуга конструирования длинных псевдослучайных рядов с «хорошими» статистическими свойствами полностью принадлежит криптографии.

Не следует думать, что они нужны лишь криптографам — картографирование Венеры стало возможным^ когда длина периода случайного ряда импульсов превысила 10. Фотографирование этой планеты нельзя было сделать потому, что она всегда закрыта плотным слоем облаков. Локация же ее с Земли затруднена обилием помех и высокими требованиями к разрешению. Поэтому зондирование выполнялось случайной последовательностью импульсов указанного периода. После 300 зондирований, на что ушло более полу- года, была получена карта, где различимы объекты размером около километра, а по высоте разрешение получено такое, какое достигнуто не везде на Земле. Генераторы псевдослучайных чисел используются очень широко в сотнях программных приложений — от конструирования ядерных реакторов и радиолокационных систем раннего обнаружения до поисков нефти и до многоканальной связи.

Можно сформировать три основных общих требования, которым должны удовлетворять криптографически стойкие генераторы псевдослучайных последовательностей и получаемые с их помощью гаммы:

* 1. Период гаммы должен быть достаточно большим для шифрования сообщений различной длины.
* 2. Гамма должна быть трудно предсказуемой. Это значит, что если известны тип генератора и кусок гаммы, то невозможно предсказать следующий за этим куском бит гаммы или предшествующий этому куску бит гаммы.
* 3. Генерирование гаммы не должно быть связано с большими техническими и организационными трудностями.

Самая важная характеристика генератора псевдослучайных чисел — это информационная длина его периода, после которого числа будут либо просто повторяться, либо их можно будет предсказать. Эта длина практически определяет возможное число ключей криптосистемы. Чем эта длина больше, тем сложнее подобрать ключ.

Второе из указанных выше требований связано со следующей проблемой: на основании чего можно сделать заключение, что гамма конкретного генератора действительно является непредсказуемой? Пока в мире нет универсальных и практически достоверных критериев для проверки этого свойства (см. обсуждение этого вопроса выше). Чтобы гамма считалась случайной и непредсказуемой, как минимум, необходимо, чтобы ее период был очень большим, а различные комбинации бит определенной длины равномерно распределялись по всей ее длине. Это требование статистически можно толковать и как сложность закона генерации псевдослучайной последовательности чисел. Если по достаточно длинному отрезку этой последовательности нельзя ни статистически, ни аналитически определить этот закон генерации, то в принципе этим можно удовлетвориться.

И, наконец, третье требование должно гарантировать возможность практической реализации генераторов псевдослучайных последовательностей с учетом требуемого быстродействия и удобства практичного использования.

Проблема генерации псевдослучайных последовательностей существует уже третье столетие. Одним из первых было предложение получать их как значения дробной части многочлена первой степени:



Если *п* пробегает значения натурального ряда чисел, то поведение *Y(ri)* выглядит весьма хаотичным. Еще Карл Якоби доказал, что при рациональном коэффициенте *а* множество *{Y(n)}* конечно, а при иррациональном — бесконечно и всюду плотно в интервале от 0 до 1. Для многочленов больших степеней такая задача была решена лишь в 1916 г. выдающимся математиком прошедшего века Германом Вейлем. Кроме того, он установил критерий равномерности распределения любой функции от натурального ряда чисел. Небезынтересно привести его в краткой формулировке.

КРИТЕРИЙ ВЕЙЛЯ. Чтобы ряд *Х, Xz, Х3,* ... был распределен равномерно в интервале от 0 до 1, необходимо и достаточно, чтобы для любой интегрируемой по Риману функции Дх) было справедливо соотношение Р{ДМ(х)) = МДх)} = 1.

Это соотношение выражает свойство, называемое эргодичностью и заключающееся в том, что среднее по реализациям псевдослучайных чисел равно среднему по всему их множеству с вероятностью 1. Таким образом, Вейль доказал, что эргодичность гарантирует отсутствие экзотичности в поведении последовательности *Хп.*

Однако эти результаты далеки от практики получения псевдослучайных рядов чисел. Дело в том, что теорема Якоби относится к действительным числам х, которые не могут быть использованы при вычислениях, потому что иррациональные действительные числа требуют для своей записи бесконечное число знаков. Попытки замены настоящего иррационального числа его приближением на ЭВМ для генерации псевдослучайной последовательности опасны, так как получаемые последовательности оканчиваются циклами с коротким периодом.

Наиболее давний вычислительный способ генерации псевдослучайных чисел на ЭВМ принадлежит Джону фон Нейману и относится к 1946 г. Этот способ базируется на том, что каждое последующее случайное число образуется возведением предыдущего в квадрат с последующим отбрасыванием цифр с обоих концов. Способ Неймана оказался ненадежным и очень быстро от него отказались.

Завершают доказательство непригодности полиномиальных и других функциональных преобразований натурального ряда чисел для получения псевдослучайных последовательностей результаты Пуанкаре. В частности, он установил, что непрерывное отображение Т области U(x) числового пространства в себя обязательно приводит к короткой цикличности траекторий Т(х) для всюду плотного в U множества точек. Ряды чисел, созданные такими методами, отягощены периодичностями.

**Генератор последовательности Фибоначчи.**Интересные классы генераторов случайных чисел неоднократно предлагались многими специалистами по целочисленной арифметике. В частности, Джордж Марсалиа и Ариф Зейман предложили класс генераторов псевдослучайных последовательностей, основанный на использовании последовательностей Фибоначчи, — в этой последовательности каждый последующий член, за исключением первых двух ее членов, равен сумме двух предыдущих:



Если эта последовательность применяется для начального заполнения массива большой длины, то, используя этот массив, можно создать генератор случайных чисел Фибоначчи с запаздыванием, где складываются не соседние, а удаленные числа. Марсалиа и Зейман предложили ввести в схему Фибоначчи «бит переноса», который может иметь начальное значение 0 или 1. Построенный на этой основе генератор «сложения с переносом» приобретает интересные свойства, на их основе можно создавать последовательности, период которых значительно больше, чем у применяемых в настоящее время конгруэнтных генераторов. По образному выражению Марсалиа, генераторы этого класса можно рассматривать как усилители случайности: «Вы берете случайное заполнение длиной в несколько тысяч бит и генерируете длинные последовательности случайных чисел».

Один из вариантов генератора последовательности Фибоначчи показан на рис. 16.1:

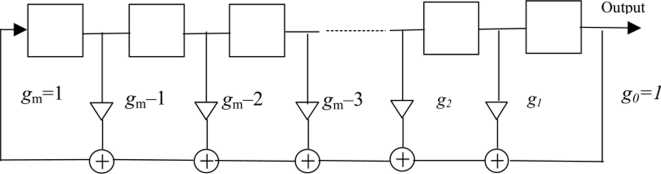


Рис. 16.1. **Генератор последовательности Фибоначчи**

Здесь квадратами обозначены разряды генератора, треугольниками — умножение на коэффициенты (на практике в зависимости от коэффициента там либо есть соединение с последующей логикой, либо его нет). Плюсы в кружках — это операция XOR: 0 + 0 = 0,

0+ 1 = 1, 1 + 1 =0.

**Рекуррентные двоичные последовательности.**Линейные последовательности максимальной длины в основном получают с помощью генераторов, использующих в качестве основных элементов TV- каскадные регистры сдвига и сумматоры по модулю 2 **(фильтр Хаффмана).**Генератор состоит из хорошо отработанных стандартных импульсных элементов и при минимальном их числе обеспечивает получение последовательности с максимальным периодом **(М-последова- тельность).**Достоинствами такого типа генератора являются фиксированная амплитуда, легко и в весьма широком диапазоне регулируемая ширина спектра сигнала, а также возможность путем небольших усложнений получать сдвинутые по шкале времени сигналы.

На рис. 16.2 представлена схема N-разрядного регистра сдвига, выходные сигналы которого после обработки при помощи специальной цифровой логической схемы снова вводятся в регистр, замыкая тем самым цепь рециркуляции. Этот регистр является базой для построения генератора псевдослучайных последовательностей. При этом необходимо выполнение следующих условий:

* - должно быть задано правило подключения сумматоров (осо, 0С2, ..., ад);
* - ао и *aN* всегда равны 1 (поэтому на схеме их можно не указывать);
* - из всех а„ /е {1, 2, ..., *N —* 1}, хотя бы одно должно иметь значение ‘Г.

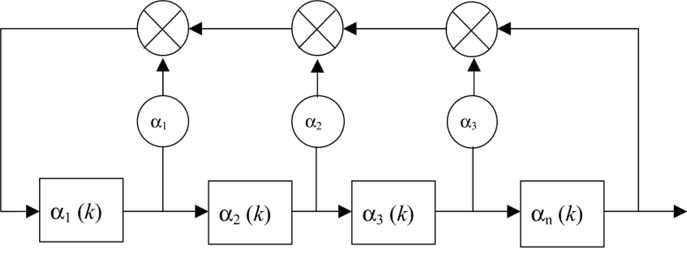
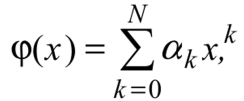


Рис. 16.2. **Фильтр Хаффмана**

Циклические свойства генератора последовательности определяются так называемым характеристическим полиномом



где cxo=av= 1, aye {0,1},*j=* 1,2, 1.

Период последовательности будет максимальным в том случае, когда многочлен ср(х) удовлетворяет условиям примитивности и неприводимости. В нахождении такого многочлена заключается основная задача синтеза генератора псевдослучайных последовательностей максимальной длины (ГППМД).

ГППМД составляют основу так называемых генераторов псевдослучайных чисел. Для этого к регистру сдвига добавляются сумматоры по модулю 2. При работе регистра сдвига на выходе этих элементах образуются М-последовательности {а\*+Л, задержанные относительно исходной последовательности {*а*>}, полученной на выходе цепи обратной связи, на определенное число тактов. Величину задержки можно регулировать в пределах от *N* + 1 до Т = 2Л - 1. При этом на выходе данного генератора получаются равномерно распределенные псевдослучайные числа.

Благодаря своему свойству цикличности, М-последовательности получили большое распространение в задачах помехоустойчивого кодирования, что дало возможность использования фильтров Хаффмана в качестве простых кодирующих и декодирующих устройств.

**Метод линейного сравнения (конгруэнтные генераторы).**Безусловно, самым популярным алгоритмом для генерирования псевдослучайных чисел является алгоритм, предложенный Д.Х. Лемером (*Lehmer*) и называемый методом линейного сравнения (конгруэнтным способом).

Этот алгоритм имеет четыре следующих параметра:

*т* модуль сравнения *т >* 0,

*а* множитель 0 < *а < т,*

*с* приращение 0 *< с < т,*

*Х()* начальное или порождающее число 0 < *Хо < т.*

Последовательность случайных чисел *{X}* получается с помощью итераций соотношения



При этом если *т, а, с* и *Хо* являются целыми, то будет получена последовательность целых чисел из диапазона 0 < *Хп < т.*

Выбор значений *а, с* и *т* оказывается очень важным в отношении разработки хорошего генератора случайных чисел. Выбор в качестве *а, с, т* иррациональных чисел, требующих для своего представления бесконечного числа знаков, практически нереализуем. Выбор почти иррациональных чисел, представленных, например, 4 байтами в формате с плавающей запятой, дает псевдослучайные последовательности с периодами всего лишь 1225 и 147 в зависимости от начального заполнения. Исследования показали, что более эффективно вести вычисления в целых числах.

Для них, в частности, было показано, что при *с —* 0, *т - 2"* наибольший период составит *т* / 4 при *а =* 3 + 8/ или *а =* 5 + 8у и нечетном начальном числе. При этом в случае достаточно больших *а* последовательность производит впечатление случайной. Дальнейшие исследования показали, что если *с* нечетно, а *а =* 1 + 4/, то период можно увеличить до числа *т =* 2". После долгих поисков числа *а*исследователи остановились на значениях *а -* 69 069 и *а -* 71 365.

Желательно иметь *т* очень большим, чтобы потенциально могли генерироваться очень длинные серии различных случайных чисел. Общим правилом здесь является выбор значения *т,* близкого к максимально допустимому для данного компьютера неотрицательному целому числу. Поэтому довольно часто значение *т*выбирается равным или почти равным значению 231.

Кроме значений *т - 2"* широко используются выборы простыхш, например, простого числа *т =* 231 - 1, известного со времен Эйлера (это число «плохо» тем, что в двоичной записи содержит лишь единицы. Однако оно может использоваться, если вычисления выполняются в десятичной арифметике).

Для иллюстрации важности выбора «хороших» параметров алгоритма рассмотрим, например, случай *а = с =* 1. Порождаемая при этом последовательность, очевидно, не будет удовлетворительной. Теперь рассмотрим значения *а -* 7, *с -* 0, *т =* 32 и *Хо =* 1. В этом случае генерируется последовательность {7, 17, 23, 1, 7, ...}, которая также, очевидно, не будет удовлетворительной. Из 32 возможных значений здесь оказываются задействованными только 4 (в данном случае говорят, что последовательность имеет период 4). Если же, оставив другие значения прежними, изменить значение *а* и положить *а =* 5, то результирующей последовательностью будет {1, 5, 25, 29, 17, 21,9, 13, 1,...} и ее период будет равен уже 8.

При реализации псевдослучайных последовательностей в компьютере или в других цифровых устройствах предлагаются три следующих критерия (вытекающие из приведенных выше основных общих требований к криптографически стойким генераторам псевдослучайных последовательностей), по которым можно оценить качество любого генератора двоичных псевдослучайных чисел.

* 1. Генерирующая функция должна быть функцией полного периода, т.е. функция должна порождать все числа от 0 до *т,* прежде чем числа начнут повторяться.
* 2. Генерируемая последовательность должна вести себя как случайная. На самом деле эта последовательность не будет случайной, поскольку генерируется детерминированным алгоритмом, но существует множество статистических тестов, которые можно использовать, чтобы оценить степень случайности поведения последовательности.
* 3. Генерирующая функция должна эффективно реализовываться в рамках 32-битовой арифметики.

Все эти три критерия могут быть удовлетворены при подходящем выборе значений *а, с* и *т.* Относительно первого критерия можно доказать, что если *т* является простым и *с =* 0, то для определенных значений *а* период генерируемой функцией последовательности оказывается равным *т* - 1, и в этой последовательности будет отсутствовать только значение 0.

В 32-битовой арифметике удобным простым значением для *т*является значение 231 - 1 (см. его недостатки выше). В этом случае генерирующая функция принимает вид

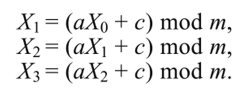


Из более чем двух миллионов возможных значений *а* только горстке множителей соответствуют функции, выдерживающие все три теста. Одним из них является значение *а =* 7 - 16 807, которое было найдено и использовано для семейства компьютеров IBM/360. Соответствующий генератор находит очень широкое применение, и поэтому он был подвергнут более тщательному анализу, чем любой другой генератор псевдослучайных чисел. Он нередко рекомендуется для статистического и имитационного моделирования различных процессов.

Преимуществом алгоритма линейного сравнения является то, что если выбрать подходящие множитель и модуль сравнения, то генерируемая последовательность чисел оказывается статистически неотличимой от последовательности чисел, выбираемых случайно (но безвозвратно) из множества чисел 1, 2, ..., *т* - 1. Но в самом алгоритме нет ничего случайного вообще, кроме выбора начального значения Xq. Если это значение выбрано, остальные числа последовательности определяются им однозначно. Это оказывается очень важным с позиций криптоанализа.

Если противник знает, что используется алгоритм линейного сравнения, и если к тому же ему известны параметры алгоритма (например, *а* = 75, *с* = 0, *т* = 231 - 1), то, открыв всего одно число, противник сможет получить и все последующие. Но даже если оппонент знает только то, что выбран алгоритм линейного сравнения, знания небольшой части последовательности уже достаточно для того, чтобы определить все параметры алгоритма.

Предположим, например, что противник сможет определить значения Xq***,****Х, Х2* и A3. Тогда



Эти уравнения могут быть решены относительно *а, с и т.*

Итак, хотя и удобно использовать хороший генератор псевдослучайных чисел, желательно позаботиться о том, чтобы генерируемая последовательность была в действительности невоспроизводимой, чтобы знание части последовательности не давало оппоненту возможности определить следующие элементы последовательности. Эта цель может быть достигнута целым рядом способов. Например, можно изменять поток случайных чисел, используя для этого системные часы. Один из способов на основе системных часов состоит в инициализации новой последовательности после получения каждых *N* чисел, используя для начального числа текущее значение времени (по mod *т).* А можно просто добавлять к каждому случайному числу текущее значение времени (по mod *т).*