УНИВЕРСИТЕТ ИТМО

М.Б. Будько
М.Ю. Будько
А.В. Гирик
В.А. Грозов

МЕТОДЫ ГЕНЕРАЦИИ И ТЕСТИРОВАНИЯ СЛУЧАЙНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

Санкт-Петербург
2019

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ
УНИВЕРСИТЕТ ИТМО

М.Б. Будько
М.Ю. Будько
А.В. Гирик
В.А. Грозов

МЕТОДЫ ГЕНЕРАЦИИ И ТЕСТИРОВАНИЯ СЛУЧАЙНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

УЧЕБНОЕ ПОСОБИЕ

РЕКОМЕНДОВАНО К ИСПОЛЬЗОВАНИЮ В УНИВЕРСИТЕТЕ ИТМО
по направлению подготовки 10.03.01, 10.04.01 в качестве
учебно-методического пособия для реализации
основных профессиональных образовательных программ
высшего образования бакалавриата и магистратуры

Санкт-Петербург
2019

Будько М.Б., Будько М.Ю., Гирик А.В., Грозов В.А. Методы генерации и тестирования случайных последовательностей – СПб: Университет ИТМО, 2019. – 70 с.

Рецензенты:
Швед Виктор Григорьевич, д.т.н., профессор, старший научный сотрудник, ЧОУ ДПО «Учебный центр «СпецПроект»

В предлагаемом пособии изложены основные сведения о случайных и псевдослучайных последовательностях, их использовании, получении и тестировании. Особое внимание уделено различным типам генераторов, проверке качества выходных последовательностей и возможностям их использования в целях криптографической защиты данных. Пособие предназначено, прежде всего, для обучающихся по направлению «Информационная безопасность» (10.03.01 и 10.04.01), изучающих дисциплины, связанные с информационной безопасностью и, в частности, криптографической защитой информации. Оно также может быть полезно выпускникам при написании выпускных квалификационных работ.

Содержание

Введение

Большинство явлений, процессов, объектов, с которыми человек встречается в жизни, имеют случайную природу. Для их адекватного описания, изучения и моделирования недостаточно применять детерминированные методы (полностью определенные некоторым алгоритмом), поэтому последовательности случайных чисел и производящие их устройства и алгоритмы (генераторы случайных последовательностей) находят широкое применение в науке, технике, связи, информационных технологиях. Особую роль случайные последовательности играют в такой области, как обеспечение информационной безопасности. Одним из наиболее эффективных и перспективных подходов к решению проблемы защиты информации является применение криптографических методов, в которых генераторы случайных последовательностей часто являются ключевыми компонентами, во многом определяя их надежность. В предлагаемом пособии изложены основные сведения о случайных последовательностях, их использовании, получении и тестировании.

Пособие содержит 5 разделов, приложение и список рекомендуемой литературы. Каждый раздел сопровождается вопросами для контроля усвоения изученного материала. Первый раздел содержит сведения о сферах применения случайных чисел и способах их получения. Второй и третий разделы посвящены соответственно генераторам истинно случайных и псевдослучайных последовательностей. Рассматриваются общие принципы работы генераторов, приводятся их классификация, основные методы генерации, характеристики, а также примеры реализаций. Для генераторов истинно случайных последовательностей обсуждается процесс постобработки. Четвертый раздел включает определения криптостойких генераторов псевдослучайных последовательностей, требования к таким генераторам, их основные типы. В пятом разделе показаны основные подходы к проверке качества генераторов случайных последовательностей. Кратко описаны наиболее распространенные наборы тестов. Подробно рассмотрен известный пакет статистических тестов NIST STS. Приведены сведения о некоторых новых средствах тестирования. В приложении приводится необходимый материал из теории вероятностей, математической статистики и теории конечных полей.

Пособие предназначено для обучающихся по направлению «Информационная безопасность» (10.03.01 и 10.04.01), изучающих дисциплины, связанные с информационной безопасностью и, в частности, с применением криптографических методов защиты информации.

Освоение изложенного в пособии материала требует знания основ теории вероятностей, теории конечных полей, а также базового представления об

алгоритмах шифрования. Разделы 1 –3 предназначены для получения начальных сведений о случайных последовательностях и методах их генерации. Они адресованы в первую очередь обучающимся бакалавриата. Разделы 4 и 5 более целесообразно использовать при обучении магистрантов, специализирующихся в области криптографии.

Раздел 1. Случайные последовательности. Области применения и способы генерации

1.1 Случайные последовательности и их применение

Большинство из окружающих нас объектов, явлений и происходящих процессов имеют случайную природу. Для адекватного описания, изучения и моделирования часто оказывается недостаточно детерминированных подходов, поэтому закономерно привлечение стохастических (т.е. имеющих случайный характер) методов решения разнообразных задач. В связи с этим случайные числа, последовательности таких чисел и производящие их генераторы находят все более широкое применение в науке, технике, связи, различных информационных технологиях, а также во многих аспектах повседневной жизни [1-13].

Исторически случайные числа начали использоваться для проведения выборочных наблюдений вместо непрерывных. Случайные числа применяются при решении сложных вычислительных задач и реализации вычислительных методов (например, метод Монте-Карло). Развитие ЭВМ, с одной стороны, расширило круг задач, использующих случайные числа, а с другой – предъявило высокие требования к качеству их генерации. Со временем случайные числа стали играть важнейшую роль в информатике, распределенных вычислениях, криптографии и других областях.

Последовательность чисел называется случайной, если воспроизвести ее, зная алгоритм и все исходные данные, не представляется возможным (дважды запустив генератор в тех же условиях, мы получим разные последовательности). Но компьютерные системы детерминированы, т.е. для них характерен строго определенный набор состояний (количество таких состояний может быть весьма большим, но конечным). Это приводит к тому, что генерируемые ими последовательности будут периодичны и воспроизводимы – такие последовательности называются псевдослучайными. Как известно, все периодическое является в той или иной степени предсказуемым, т.е. неслучайным. Получение истинно случайных последовательностей достаточно трудоемко. К тому же для их создания подходит далеко не каждый физический или информационный процесс.

Случайные последовательности находят применение в самых разных сферах человеческой деятельности. Ниже приведен перечень самых известных направлений, в которых случайные последовательности применяются наиболее интенсивно.

1. **Криптография.** Криптографические методы являются базовыми в обеспечении информационной безопасности. В криптографии случайные

последовательности играют определяющую роль. Они используются, в частности, для получения ключевой последовательности используемого алгоритма шифрования, для генерации гаммы поточных шифров, а также для выработки векторов инициализации (блочных шифров).

2. **Другие направления защиты информации.** Случайные последовательности незаменимы при формировании паролей и пользовательских ключей (хороший пароль представляет собой короткую последовательность случайных символов). Кроме этого, они могут использоваться для внесения неопределенности в результаты работы различных алгоритмов защиты информации, а также в длительность выполнения шагов алгоритмов для защиты от утечек по побочным каналам. Они также необходимы при формировании случайных запросов при аутентификации и решении многих других задач.

3. Тестирование алгоритмов. Важной задачей является проверка правильности работы программ. Тестирование – достаточно долгий и трудоемкий процесс. Для его осуществления требуется большой объем входных данных. Использование генераторов случайных чисел повышает эффективность тестирования и позволяет экономить время.

4. **Сетевые протоколы.** Случайные последовательности могут использоваться, например, в качестве сессионных ключей, а также для выработки случайных параметров протокола, что обеспечивает уникальность его различных реализаций.

5. **Математическое и имитационное моделирование.** При моделировании сложных физических, технологических и социально-экономических систем и процессов обойтись без применения источников случайности не представляется возможным.

6. **Математическая статистика.** Математическая статистика изучает приближенные методы сбора и анализа данных по результатам эксперимента для выявления существующих закономерностей, т.е. нахождение законов распределения случайных величин и их числовых характеристик. Необходимой составляющей выборочных методов является формирование представительных выборок из генеральной совокупности с использованием случайных чисел.

Также случайные последовательности часто используются в теории чисел; статистической физике; прогнозировании; методах вычислений (в том числе методе Монте-Карло); теории управления; информационных технологиях для банковских, платежных, торговых систем; помехоустойчивом кодировании; автономном и встроенном диагностировании компонентов компьютерных систем; модуляции радиосигналов; в контроле хода выполнения программ с использованием сторожевых процессоров; индустрии игр и лотереях.

Одной из сфер деятельности, в которых случайные последовательности играют важную роль, является защита информации.

Случайные последовательности, применяемые в различных аспектах защиты информации, используются для решения следующих задач:

- генерация гаммирующих последовательностей при поточном шифровании информации по схеме, наиболее близкой к схеме абсолютно стойкого шифра;
- формирование векторов инициализации для блочных шифров, работающих в режиме обратной связи;
- получение начальных значений для программ генерации некоторых параметров в асимметричных криптосистемах;
- формирование пользовательских ключей и паролей;
- формирование ключевой информации, на секретности и качестве которой основывается стойкость большинства криптоалгоритмов;
- формирование случайных запросов при реализации большого числа криптографических протоколов, например, протоколов выработки общего секретного ключа, разделения секрета, привязки к биту, аутентификации, электронной подписи и др.;
- внесение неопределенности в работу средств защиты, например, при реализации концепции вероятностного шифрования, при котором одному и тому же исходному тексту при одном и том же ключе соответствует огромное множество шифротекстов;
- выполнение статистического тестирования;
- формирование затемняющих множителей при слепом шифровании (протокол слепой подписи).

Особое место занимает использование случайных чисел в криптографии – одном из самых мощных и эффективных методов защиты информации. В криптографии ключевую роль играют последовательности битов – двоичные последовательности, состоящих из случайно разделенных значений «0» и «1». Именно такие последовательности и будут называться в этом пособии «случайными».

Для ряда криптографических преобразований используют случайные первичные состояния либо целые последовательности. Следовательно, стойкость криптоалгоритма, использующего такие состояния или последовательности, напрямую зависит от алгоритма генерации случайных чисел и последовательностей, точнее от степени случайности выходных последовательностей.

Широта и важность областей применения случайных последовательностей, их определяющая роль в обеспечении высокого уровня защиты информации обуславливают актуальность их изучения.

Процесс генерации случайных чисел является основной частью многих криптографических операций. Например, криптографические ключи должны выбираться настолько случайно, насколько это возможно, чтобы на практике нельзя было воспроизвести их значения. Криптографически стойкие генераторы случайных чисел должны выдавать данные, которые невозможно предугадать с вероятностью выше 0,5; это означает, что любой метод предсказания очередного выходного бита не должен действовать эффективнее, чем просто случайное угадывание.

1.2 Способы генерации псевдослучайных последовательностей

В настоящее время существует большое количество способов генерации последовательностей, обладающих той или иной степенью случайности [1-4, 7-9, 14-15]. Однако на практике большинство из таких генераторов производят последовательности, свойства которых не удовлетворяют требованиям случайности. Один из самых распространенных примеров этого – генераторы псевдослучайных чисел, встроенные в стандартные библиотеки многих языков программирования (такие, как, например, функция стандартной библиотеки языка C *rand*()). Часто в числах, сгенерированных с помощью подобных функций, прослеживаются явные закономерности. Например, полученные числа в одном и том же сеансе с течением времени монотонно возрастают, что прямо противоречит требованиям, предъявляемым к свойствам случайных (и псевдослучайных) последовательностей. Для широко известных и распространенных линейных конгруэнтных генераторов по четырем известным сгенерированным числам также можно предсказать дальнейшие значения.

Большинство криптографических приложений используют генераторы случайных чисел для создания ключей, с помощью которых шифруется и расшифровывается нужная информация. Однако часто именно применяемые в них генераторы являются самым слабым местом в системах шифрования. Дело в том, что программные генераторы полностью детерминированы. Обычно они используют различные сложные функции для вычисления псевдослучайных чисел. Соответственно, последовательности, полученные в результате работы таких генераторов, являются в той или иной степени предсказуемыми и воспроизводимыми и не подходят, например, для использования в криптографических приложениях. Необходимо отметить, что в некоторых случаях возможность воспроизвести случайную последовательность является полезной (например, при тестировании алгоритмов разработчиком). Тем не менее, последовательность не должна обладать свойствами, которые позволили бы злонамеренному криптоаналитику восстановить ее в процессе анализа работы защищенного приложения или протокола.

Существует табличный способ генерации случайных последовательностей. Он заключается в том, что случайные числа оформлены в виде таблицы, бумажной

или электронной, которая хранится в оперативной памяти или на внешнем носителе. Один из вариантов таблицы случайных чисел и способа их выбора описан в ГОСТ Р ИСО 24153-2012 (Статистические методы. Процедуры рандомизации и отбора случайной выборки). Достоинство этого способа состоит в том, что с его помощью можно воспроизводить неоднократно одну и ту же последовательность псевдослучайных чисел. Однако серьезным недостатком, фактически не допускающим применения таких генераторов при решении практических задач, является то, что запас доступных чисел ограничен. Также при таком подходе возможно неэффективное использование вычислительных ресурсов компьютера (например, из-за необходимости хранить таблицу или ее части в оперативной памяти или обращаться к внешней памяти). В настоящее время такой способ генерации используется достаточно редко.

Среди генераторов псевдослучайных последовательностей (ГПСЧ), получивших широкое распространение и применимых при решении задач с серьезными требованиями к качеству сгенерированной последовательности, различают аппаратные, программные и программно-аппаратные (смешанные).

Аппаратный генератор случайных чисел – это устройство, которое генерирует последовательность случайных чисел на основе измеряемых, хаотически изменяющихся параметров протекающего физического процесса. При аппаратном способе генерации случайные числа являются прямым или побочным продуктом измерений некоторой физической величины, служащей надежным источником энтропии. Обычно это процессы, протекающие в неживой природе. Теоретически такие процессы абсолютно непредсказуемы, однако на практике полученные таким образом случайные числа приходится подвергать проверке с помощью специальных статистических тестов. Несмотря на лучшие статистические свойства и, соответственно, более высокую степень случайности, аппаратным генераторам присущи следующие недостатки:

- потенциально высокие временные и материальные затраты на конструирование, установку и настройку по сравнению с программными ГПСЧ;
- более низкая скорость генерации случайных чисел, чем при программной реализации ГПСЧ [14, 15];
- невозможность воспроизведения ранее сгенерированной последовательности чисел (что в некоторых случаях является нежелательным).

Программные (алгоритмические) генераторы (генераторы псевдослучайных последовательностей) основаны на детерминированных алгоритмах. У полученных таким образом последовательностей всегда существует период (пусть иногда и очень большой), а также наблюдаются и другие отклонения от случайности. Любой ГПСЧ с ограниченными ресурсами рано или поздно

зацикливается – начинает повторять одну и ту же последовательность чисел. Период ГПСЧ зависит от типа генератора и его параметров [1, 4, 7, 9]. Если порождаемая последовательность ГПСЧ имеет слишком короткий период, то такой ГПСЧ становится непригодным для многих практических приложений.

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

- слишком короткий период;
- последовательные значения не являются независимыми;
- некоторые биты «менее случайны», чем другие;
- неравномерное распределение;
- обратимость.

Фактически, результат работы таких генераторов не является случайной последовательностью. Тем не менее, к последовательностям, производимым программными генераторами, предъявляются определенные требования, поскольку они должны в какой-то степени имитировать случайные последовательности. В частности, период таких последовательностей должен быть достаточно большим, чтобы при генерации последовательности требуемой длины не возникало повторений. В отличие от аппаратных генераторов, программные генераторы способны воспроизвести ранее сгенерированную последовательность, что в некоторых случаях является бесспорным преимуществом.

Программно-аппаратные генераторы. Такой генератор может формировать поток случайных шумов, которые затем преобразуются в числа. Также возможен вариант, когда «зерно» (т.е. некие входные данные алгоритма шифрования) генерируется с помощью аппаратного генератора (поскольку ее размер достаточно небольшой, и, соответственно, ее получение не требует больших затрат времени и ресурсов), а итоговая последовательность – с помощью программного.

К программно-аппаратным генераторам случайных чисел могут относиться, например, устройства компьютера. В частности, источником случайной последовательности могут быть шумы устройств компьютера (например, процессора), системное время, временные интервалы между нажатиями клавиш, движения мыши и так далее. Как правило, последовательности, получившиеся в результате таких процессов, нуждаются в постобработке. К тому же скорость их получения является достаточно низкой (особенно при генерации последовательностей достаточно больших объемов). К таким генераторам можно отнести, в частности, псевдоустройства /dev/random и /dev/urandom ОС Linux.

В последующих разделах описанные выше подходы к генерации случайных последовательностей, а также процесс статистического тестирования генераторов случайных последовательностей будут рассмотрены подробнее.

Последовательность называется истинно случайной (ИСП), если ее нельзя воспроизвести. Это означает, что если запустить генератор истинно случайных последовательностей дважды при одном и том же входе, то на его выходе получатся разные случайные последовательности. Основная трудность состоит в том, чтобы суметь отличить случайную последовательность от неслучайной.

Однако на практике далеко не всегда можно непосредственно использовать выходные данные источников истинно случайных чисел. Поэтому обычно приходится использовать так называемые псевдослучайные последовательности. Псевдослучайная последовательность (ПСП) – это последовательность, состоящая из псевдослучайных двоичных чисел, получаемых с помощью заданного детерминированного алгоритма, но применяемых в качестве случайных. При этом обычно алгоритмы получения ПСП используют специальное случайное начальное значение, или «зерно» (*seed*). Для того чтобы ПСП могли использоваться в качестве случайных последовательностей, они должны по статистическим свойствам быть близки к ИСП.

В табл. 1.1 приведено сравнение основных характеристик обоих типов СП.

Таблица 1.1 – Характеристики ИСП и ПСП

Характеристика	Случайные последовательности	Псевдослучайные последовательности
Отсутствие периодичности	да	нет
Непредсказуемость	да	условная
Независимость значений	да	условная
Уровень криптостойкости	высокий	условный
Скорость генерации	низкая	высокая
Воспроизводимость	нет	да
Простота генерации	нет	да
Стоимость генерации	высокая	низкая

Мы знаем, что на микроуровне случайность существует (квантовая механика), но неизвестно, сохраняется ли эта случайность при переходе на макроуровень. Дополнительное свойство случайной последовательности заключается в том, что случайная последовательность не может быть сжата [16].

Требования к качественному генератору случайных чисел [3]:

1. Непредсказуемость результатов работы: при неизвестном ключе/начальном состоянии генератора на основе известной конечной части ПСП невозможно определить как ее последующий элемент (прямая непредсказуемость, или непредсказуемость вправо), так и предыдущий (обратная непредсказуемость, непредсказуемость влево);
2. Неотличимость статистических свойств генерируемых ПСП от аналогичных свойств истинно случайной последовательности;
3. Большой период последовательности;
4. Возможность эффективной аппаратной и программной реализации.

На практике добиться выполнения всех этих условий, как правило, не представляется возможным. Более того, часто эти условия являются взаимоисключающими. Поэтому приходится искать баланс между ними и в первую очередь стремиться к выполнению того, что является наиболее важным в контексте решаемой задачи.

Часто наилучшие результаты получаются в случае комбинирования разных способов генерации случайных последовательностей. Например, начальная информация может быть получена при помощи аппаратного генератора, а сама итоговая последовательность – с помощью программного, получившего на вход начальные данные с аппаратного генератора.

Вопросы для самоконтроля

1. Дайте определение случайной последовательности.
2. Перечислите основные области применения случайных последовательностей.
3. Как случайные последовательности используются при решении задач обеспечения информационной безопасности?
4. Охарактеризуйте основные способы генерации случайных последовательностей.
5. Дайте определения истинно случайных и псевдослучайных последовательностей. Перечислите основные достоинства и недостатки случайных и псевдослучайных последовательностей.
6. Сформулируйте перечень требований к качественному генератору случайных чисел.

Раздел 2. Генераторы истинно случайных последовательностей

Истинно случайные последовательности (ИСП) – это представленные в виде последовательности случайные числа. Случайные числа являются реализацией некоторой случайной величины. Случайная величина – это функция над пространством элементарных событий, принимающая вещественные значения с некоторыми вероятностями. Таким образом, истинно случайные последовательности – это последовательности статистически независимых друг от друга величин, принимающих в результате опыта одно из множества заранее непредсказуемых значений. Связь между значениями случайной величины и соответствующими вероятностями описывается законом распределения, который задается с помощью функции распределения или связанной с ней функцией плотности вероятности. Истинно случайные последовательности должны обладать равномерным распределением.

Генераторы истинно случайных последовательностей – это устройства, которые для получения случайных последовательностей используют объективно существующую случайность физических процессов, происходящих как на макроуровне, так и на микроуровне [14, 15]. Как правило, такие генераторы являются аппаратными, либо программно-аппаратными.

Потребность в ГСП возникает в тех случаях, когда требуется очень высокая степень случайности, недостижимая при использовании ГПСП (квантовая криптография, начальное состояние ключа для блочных шифров).

2.1 Принципы работы

В аппаратных генераторах случайных последовательностей для генерации случайных чисел используются такие источники энтропии, как:

- тепловой и электрический шум;
- квантовые процессы;
- радиоактивный распад;
- космическое излучение;
- различные механические, оптические и фотоэлектрические явления.

Конкретными примерами источников энтропии, подходящих для генерации истинно случайных последовательностей, могут быть:

- временные интервалы между выбросами частиц при радиоактивном распаде;
- тепловой шум полупроводникового диода или резистора;
- состояния спутанности фотонов;
- квантовый шум лазеров;

- дробовой шум;
- нестабильности частоты осцилляторов.

Программно-аппаратные генераторы истинно случайных последовательностей основаны на случайностях, присущих работе компьютерного оборудования, таких, как показания системных часов, уровень загрузки процессора, задержки между прибытиями сетевых пакетов, интервалы времени между срабатываниями мыши или клавиатуры, содержимое буферов ввода/вывода, шумы процессоров или других устройств.

Реальное применение ГСП может быть затруднено такими их особенностями, как:

- низкая скорость работы;
- сложность повторного воспроизведения, дублирования и взаимодействия с процессором;
- отклонения и корреляции в получаемых последовательностях, связанные с систематическими ошибками в ходе измерений или наличием волновых или других периодических (неслучайных) составляющих, выявляемые при статистическом тестировании.

Результат работы ГСП может потребовать дополнительной обработки (так называемая постобработка).

Полученная случайная последовательность может использоваться непосредственно или быть входной для генератора псевдослучайных последовательностей.

Особое место среди источников случайных данных занимают процессы, описываемые в рамках квантовой физики. Их вероятностная природа делает теоретически возможным получение истинно случайных последовательностей. Существуют и разрабатываются квантовые генераторы истинно случайных чисел, основанные на явлениях радиоактивного распада, запутанных квантовых состояниях, лазерном квантовом шуме, квантовых флуктуациях в вакууме, процессах эмиссии и детектирования фотонов.

Физические явления различной природы принято называть «случайным шумом» (белым шумом), если они представляют собой беспорядочные колебания.

Известный пример случайного шума – тепловой шум, или шум Джонсона [15]. Это колебания напряжения, измеренного для любого материала, обладающего электрическим сопротивлением и находящегося при температуре выше абсолютного нуля. Причиной таких колебаний является тепловое движение носителей электрического заряда, имеющее случайный характер. Следует, однако, заметить, что указанное напряжение в реальности не является полностью случайным, поскольку существуют определенные корреляции носителей в проводниках, вызывающие корреляции в движениях электрических зарядов.

Туннельный эффект Зенера [14, 15], наблюдаемый в полупроводниковых стабилитронах (специальных диодах Зенера, которые способны работать в условиях обратного смещения в зоне пробоя), вызывает случайные скачки напряжения при переходах носителей через квантовый барьер (так называемый «розовый шум»). При этом эффект Зенера не изолирован полностью в физических устройствах от других эффектов. К тому же для названных процессов в сопротивлениях и стабилитронах характерен эффект памяти: мгновенное напряжение на устройстве зависит от напряжения в недавнем прошлом. Это приводит к корреляции полученных таким образом чисел и не дает права назвать такую последовательность истинно случайной.

Можно назвать и другие популярные источники шума, например, пробой база – эмиттер в биполярных транзисторах, фазовый шум лазера, хаотический шум и др. Однако общая проблема для всех этих источников шума состоит в том, что порождаемую ими случайность невозможно абсолютно точно проконтролировать при изготовлении соответствующего устройства или измерить. Значения напряжения (например, для шума Джонсона) могут быть очень малы, что требует существенного усиления перед преобразованием в цифровую форму. Это добавляет дополнительные отклонения из-за ограниченной полосы пропускания усилителя и нелинейности коэффициента усиления. При быстром переключении двоичной логики, которое используется в схеме генератора случайных последовательностей, возникают сильные электромагнитные помехи, из-за которых находящиеся вблизи генераторы (особенно расположенные на одном чипе), обычно взаимно синхронизируются, что приводит к резкому падению общей энтропии. Существует также опасность криптографических атак на шумовые генераторы истинно случайных последовательностей путем воздействия на них высокочувствительных усилителей.

Основная идея построения генератора истинно случайных последовательностей, базирующихся на источниках шума, состоит в следующем. Случайное аналоговое напряжение, поступающее от источника шума, периодически дискретизируется, усиливается и подается на компаратор для сравнивается с заранее выбранным порогом. При превышении этого порога генерируется значение «1», в обратном случае генерируется «0». Порог может быть установлен таким образом, чтобы вероятности появления «1» и «0» будут примерно равны. Процедура настройки порога оказывается сложной, отнимающей много времени и может вызвать заметное искажение показаний генератора шума.

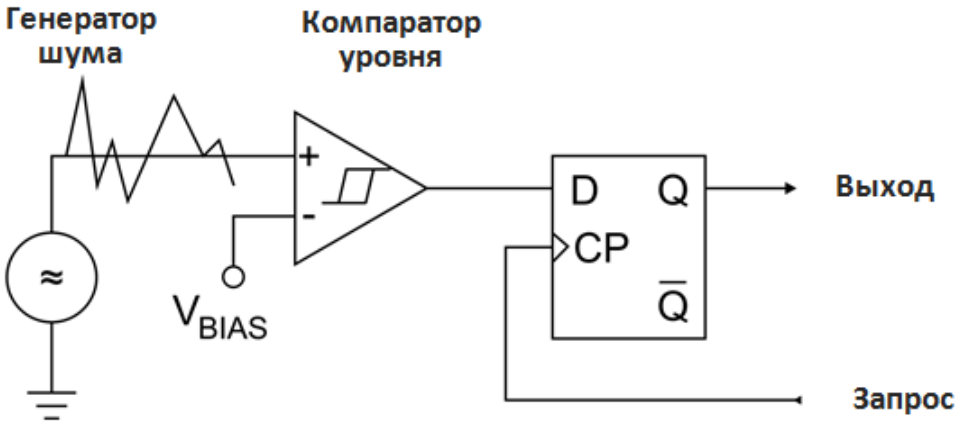

Рисунок 2.1 – Схема генератора истинно случайных последовательностей, основанного на источниках шума.

Для приведенной базовой схемы предлагаются различные модификации, направленные на улучшение степени случайности выходных данных генератора, и в особенности – на уменьшение смещения, которое содержится в необработанном потоке шума. Примером одного из таких решений служит генератор Баджини-Буччи [15, 17].

В целом надежность любого генератора случайных последовательностей, основанных на шуме, зависит от следующих факторов:

- степень случайности используемого источника шума;
- влияние процедур выборки и оцифровки сигналов;
- необходимость использования детерминированной постобработки.

В итоге по указанным причинам доказательство надежности ГСП на основе шума становится практически невозможным.

2.2 Классификация генераторов истинно случайных последовательностей

Генераторы случайных последовательностей, основанные на хаотических процессах

Этот тип генераторов использует состояние детерминированного хаоса динамических систем [15]. Отчасти такое применение неслучайной системы в качестве источника истинно случайных чисел объясняется мнением о близости хаоса и случайности. Другой причиной является возможность получения в некоторых хаотических системах макроскопических уровней «шума». Это позволяет выполнять генерацию случайных чисел с помощью методов шумовых ГСП.

Наиболее разработанными хаотическими системами для генерации случайных чисел являются оптические, электрические и оптоэлектрические системы.

Так, лазеры различными способами могут быть приведены в состояние хаотической флуктуации мощности при помощи использования различных механизмов. Например, высокую скорость генерации последовательностей (до 300 Гбит/с) обеспечивает простая установка на основе лазера с внутренней обратной связью (рис. 2.2), позволяющая с помощью фотодиода считывать амплитуды интенсивности лазера с дискретизацией быстрым аналого-цифровым преобразователем и последующей обработкой путем выполнения дифференцирования высокого порядка.

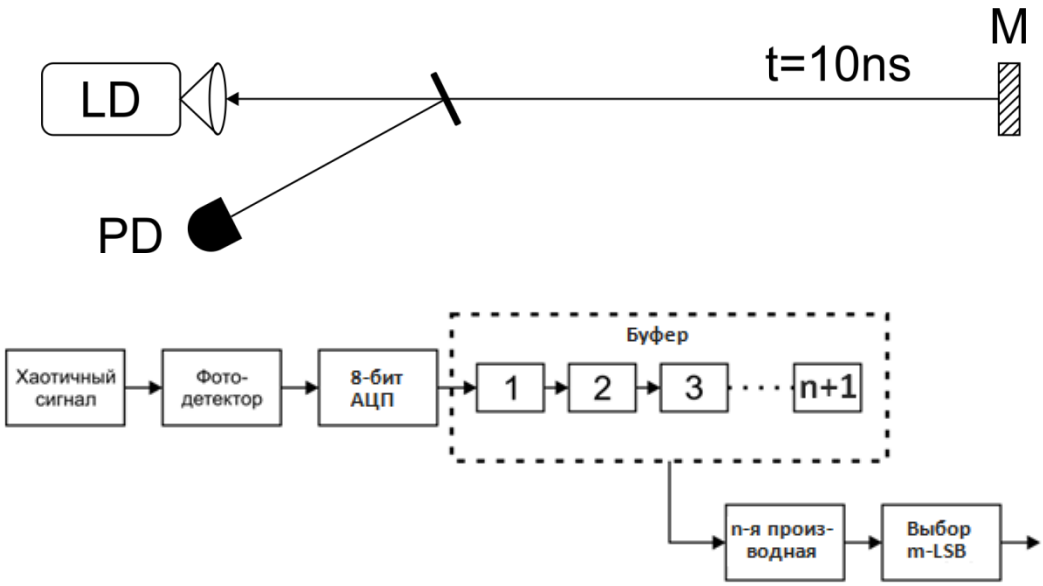

Рисунок 2.2 – Схема хаотического генератора на основе лазера с внутренней обратной связью (LSB – наименее значащие биты).

Лазеры позволяют реализовывать очень быстрые хаотические системы, а благодаря возможности построения на чипе/микросхеме/кристалле миниатюрных лазеров, резонаторов и различных активных и пассивных оптических элементов, эти генераторы могут быть полностью интегрированы и характеризуются низким энергопотреблением.

Генераторы свободных колебаний

Кольцевые генераторы

В том случае, когда выходной сигнал инвертора подается на его вход, цепь превращается в так называемый свободный осциллятор, или генератор свободных

колебаний (рис. 2.3). Частота его колебаний определяется внутренними запаздываниями и паразитными емкостями.

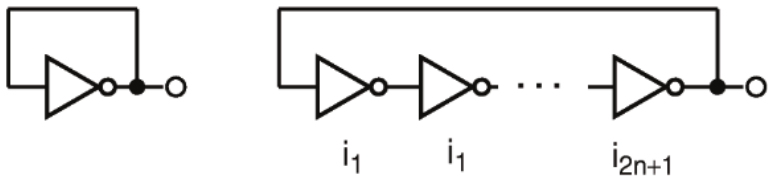

Рисунок 2.3 – Схема быстрого (слева) и медленного (справа) свободного осциллятора.

Инвертирующий вентиль представляет собой инверсный усилитель высокой мощности. Особенность возникающих в такой схеме колебаний заключается в том, что они возникают в цепи с отрицательной обратной связью (сдвиг фазы 180 градусов), в то время как обычно отрицательная обратная связь приводит к стабилизации. Причина заключается в том, что при анализе системы предполагается, что усиление является бесконечным. Однако, поскольку в реальных условиях такое усиление недостижимо, цепь может перейти в некоторое промежуточное состояние, с нулевыми колебаниями или с колебаниями очень малой амплитуды. Для поддержания колебаний можно добавить некоторое реактивное сопротивление в петлю обратной связи, чтобы произвести сдвиг фазы, отличный от ±180 градусов. Такая же функция может быть обеспечена с рассеянным реактивным сопротивлением. Из-за сложного механизма свободных колебаний их частота, как правило, весьма чувствительна к изменению напряжения питания и температуры, но эти изменения в сравнении с частотой колебаний происходят медленно. С другой стороны, электрический шум, присутствующий на входе, прибавляется к сигналу, поданному назад от выхода, и после того, как он подвергается серьезному усилению, вызывает очень сильное случайное колебание (дрожание) частоты и фазы колебаний. В этом смысле генераторы случайных чисел, основанные на генераторах свободных колебаний, можно рассматривать как частный случай генераторов на основе шума. Поскольку шум каждого такого контура является индивидуальным, разумно предположить, что несколько генераторов даже на одной микросхеме имеют разные частоты и что их взаимные фазы случайны по времени. Если несколько таких генераторов расположены близко друг к другу (например, на одной микросхеме), они имеют тенденцию синхронизироваться через электромагнитное взаимодействие, чему способствует высокий коэффициент усиления свободных осцилляторов. Этот эффект является основной проблемой, присущей генераторам свободного хода, и может отрицательно сказаться на надежности генератора.

Другой важной проблемой генераторов на свободных осцилляторах является то, что выходная амплитуда генератора зависит от паразитных реактивностей и задержек в цепи.

В ГОСТ 28640-2012 (Статистические методы. Генерация случайных чисел) в качестве рекомендуемого источника физических случайных чисел указан электрический шум диода. Отмечается, что шумовой сигнал диода достаточно велик вследствие эффекта лавинного нарастания заряда. Названы такие методы преобразования шумового сигнала в цифровую форму, как:

1. Аналогово-цифровое преобразование.
2. Наблюдение последовательности импульсов с определением количества импульсов в единицу времени.
3. Наблюдение последовательности импульсов с определением интервала времени между последовательными импульсами.

Квантовые генераторы случайных последовательностей

Основу физического описания нашего мира составляют законы квантовой физики. С точки зрения поисков надежных источников энтропии важнейшей особенностью квантовых процессов является не вполне детерминированное математическое описание движения частиц. Случайность внутренне присуща квантовым процессам, которые естественно рассматривать как хороший источник случайных чисел.

Действительно, все рассмотренные выше системы, используемые как ГСП, подчиняются законам классической физики. Случайность их поведения вызвана лишь неопределенностью начальных состояний, что теоретически делает эти системы предсказуемыми. Поэтому только генераторы, работающие на квантовых принципах, можно строго обоснованно считать производящими истинно случайные значения.

Квантовыми генераторами принято называть такие устройства, которые используют один действительно случайный квантовый эффект, который возможно воспроизводить многократно для получения случайных значений таким образом, чтобы перед каждым измерением система возвращалась к тем же начальным условиям. Важно отметить, что при одинаковых начальных значениях и одном и том же способе измерения в соответствии с принципами квантовой физики будут получены различные результаты.

Приведем основные сведения для объяснения принципов работы квантовых генераторов.

Базовым понятием является кубит (*qubit*) [18-20], квантовый аналог обычного бита. Кубит определяют как квантовую систему, которая может находиться в двух состояниях: 0 и 1. В области элементарных частиц пример такой системы – электрон (имеет два возможных направления спина) или фотон (две

возможные поляризации). Идея квантовых вычислений основана на таких эффектах квантовой механики, как квантовая суперпозиция и квантовый параллелизм, и использовании квантовых систем из двухуровневых квантовых элементов (кубитов). Система из L кубитов имеет 2^L линейно независимых состояний и может выполнять параллельно 2^L операций. В рамках квантовой теории поля можно доказать, что при измерении состояния кубита его можно обнаружить в одном из двух указанных возможных состояний, а это означает, что из кубита можно извлечь один бит информации.

В квантовых генераторах случайных чисел часто используются фотоны, т.к. их легко создавать, обнаруживать и манипулировать ими.

В качестве иллюстрации можно привести схему ГСП, основанного на прохождении фотона, имеющего круговую поляризацию, через светоделительную пластину (рис. 2.4) [15]. Такая система с равной вероятностью обеспечивает выход фотона как в вертикальном, так и в горизонтальном направлении. При неизменной системе и ее начальном состоянии результат каждый раз может быть иным, т.е. является случайным.

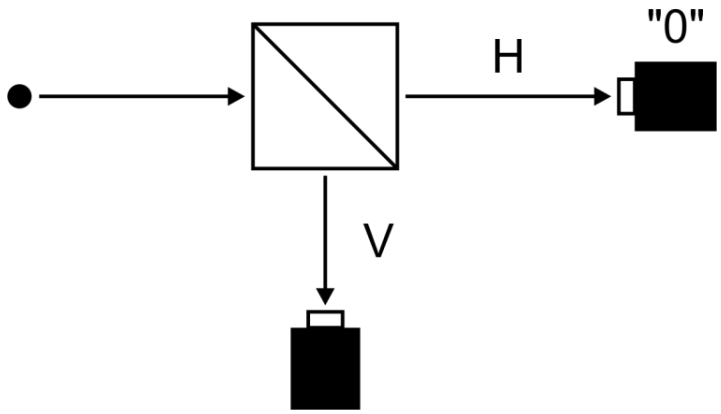

Рисунок 2.4 – Схема квантового ГСП, основанного на прохождении фотона с круговой поляризацией.

Следует также добавить, что качество квантовых ГСП не снижается из-за существующих недостатков, таких, как неидеальная поляризация, многофазное излучение, время простоя детектора и т.д. Это объясняется тем, что вносимые ими отклонения могут измеряться и оцениваться независимо от процесса генерации случайных значений.

Основная проблема практической реализации генераторов случайных чисел, базирующихся на расщеплении пучка света, состоит в том, что для этого требуются два детектора. Их изначальные различия и последующий выход из строя из-за старения или температурных эффектов оказывают непосредственное влияние на качество сгенерированных случайных чисел.

Например, если эффективность детектирования фотонов датчиками не совсем одинакова, или если светоделительная пластина не идеально делит пучок

на две равные части, то при использовании такого генератора вероятность появления единиц не будет равна вероятности появления нулей. Эту проблему можно свести к минимуму с помощью схемы расщепления пучка, в которой используется только один фотонный детектор, но при этом коэффициент расщепления пучка должен быть точно отрегулирован механически.

Другие проблемы возникают из-за времени простоя детектора и последующего импульса, что приводит к корреляциям, которые невозможно полностью устранить, но которые могут быть уменьшены ниже любого желаемого (допустимого) уровня путем применения целевой постобработки.

Генераторы случайных чисел, основанные на светоделителях, являются примером так называемого «пространственного принципа», в котором значение, принимаемое случайным битом (0 или 1), определяется местом, в котором фотон заканчивается. Дополняющий его «временной принцип» использует информацию о времени выбросов случайных фотонов, например, в прямой квантовой (или атомной) релаксации, от хорошо насыщенных лазеров и т.д.

2.3 Постобработка

При получении случайных значений с помощью как квантовых, так и других генераторов следует помнить, что непосредственно на выходе мы имеем пока только «сырые» последовательности, нуждающиеся в дополнительной обработке. Результирующая последовательность битов должна иметь распределение, максимально близкое к равномерному. Для достижения желаемого результата требуется отдельный важный этап, называемый постобработкой и реализуемый путем применения различных специальных алгоритмов (рис. 2.5).

Смысл постобработки состоит в готовности пожертвовать определенной долей битов, чтобы получить меньший, но обладающий более высокой степенью случайности набор.

Рисунок 2.5 – Схема процесса постобработки СП.

Выделяют следующие основные методы постобработки [15]:

1. Специальные простые корректоры (*ad hoc simple correctors*).
2. «Отбеливание» (*whitening*) последовательностей с помощью криптографических хэш-функций.

3. Алгоритмы экстракторов.
4. «Упругие» функции (*resilient functions*).

Иногда этап получения «сырых» случайных чисел называют экстракцией, понимая под ней преобразование физических измерений аналогового или цифрового сигнала в биты [15]. Постобработкой считается более сложный этап, выполняемый для устранения недостатков полученной последовательности (наличие корреляций, сдвигов). Однако более распространен подход, при котором нет четкой границы между этими процессами.

Хотя извлечение битов обычно осуществляется аппаратно, алгоритмы постобработки обычно настолько сложны, что они могут быть выполнены только с помощью компьютера (или микроконтроллера), хотя наиболее ценные методы постобработки достаточно просты, чтобы быть пригодными для прямой реализации в аппаратном обеспечении.

Специальные простые корректоры

Примерами таких процедур являются, например, применение операции XOR к двум или более соседним битам, полученным от одного генератора; опускание битов; использование латинского квадрата для перемешивания битов; экстрактор фон Неймана; применение операции XOR к битам последовательностей, полученных от двух или более параллельно работающих генераторов.

Так, экстрактор фон Неймана выполняет устранение смещения следующим образом: для каждой пары сгенерированных битов отбрасываются результаты вида 00 и 11, 01 принимается за 0, а 10 – за 1. Если в исходной последовательности присутствует систематический сдвиг, такой метод убирает его, но при этом теряется примерно половина битов и по меньшей мере в четыре раза падает скорость передачи битов.

Следует учитывать, что применение таких простых методов не всегда бывает эффективным и корректным. Например, если исходная строка битов коррелирована, процедура устранения смещения Неймана может даже увеличить смещение или создать другие неожиданные статистические недостатки.

Одной из самых известных техник постобработки является «отбеливание» выходных данных генератора истинно случайных чисел с помощью криптографической хэш-функции [3, 8, 9, 15]. Наиболее известными являются такие хэш-функции, как MD5, SHA-1, SHA-2, SHA-256, SHA-512. Считается, что если имеющие недостатки выходные последовательности генератора случайных чисел пропустить через хэш-функцию, то их качество значительно улучшится. На самом деле показано, что хэширование «плохого» генератора может не повысить случайность до уровня, достаточного для прохождения статистических тестов [15].

Функции экстракции

Более тщательный подход к повышению уровня случайности предлагает недавно появившаяся теория экстракторов. Экстрактор случайности – это алгоритм, который преобразует длинную слабо случайную последовательность в более короткую последовательность, обладающую почти идеальной случайностью. Проблемой таких алгоритмов является то, что для их работы необходим буфер памяти, а также мощный процессор, что замедляет скорость выходного потока битов [15].

«Упругие» функции

Еще один подход к усилению случайности путем фильтрации через некоторый детерминированный процесс – это использование упругих функций, которые были введены Сунаром, Мартином и Стинсоном [15, 21] в качестве шага постобработки для ГСП на свободных осцилляторах.

Вопросы для самоконтроля

1. Что такое истинно случайная последовательность?
2. Что собой представляет генератор истинно случайных последовательностей?
3. Перечислите наиболее известные источники случайности, встречающиеся в природе.
4. Какие генераторы случайных последовательностей, основывающиеся на хаотических процессах, применяются чаще всего?
5. Охарактеризуйте квантовые генераторы истинно случайных последовательностей.
6. В чем заключается сущность постобработки выходных данных генераторов?

Раздел 3. Генераторы псевдослучайных последовательностей

3.1 Применение генераторов псевдослучайных последовательностей

Как уже отмечалось, псевдослучайные последовательности (ПСП) являются результатом работы некоторого детерминированного алгоритма, но при этом используются для тех же целей, что и случайные последовательности. По этой причине ПСП должны быть близки по своим статистическим свойствам к истинно случайным последовательностям. Как правило, алгоритмы генерации таких последовательностей используют специальное случайное начальное значение (*seed*).

В силу особенностей генераторов истинно случайных последовательностей, которые серьезно затрудняют их использование в условиях ограниченности вычислительных ресурсов и необходимости работы в режиме, близком к режиму реального времени, существует необходимость разработки и применения генераторов псевдослучайных последовательностей, которые производят последовательности, максимально близкие по своим свойствам к истинно случайным.

Общая структура генераторов псевдослучайных последовательностей включает следующие основные компоненты:

1. Источник энтропии.
2. Блок хранения внутреннего состояния генератора.
3. Блок генерации очередного значения.
4. Блок перехода генератора на следующий шаг.

В настоящее время существует большое количество самых разных генераторов псевдослучайных последовательностей [1-4, 7, 9]. К наиболее популярным относятся:

- линейные и нелинейные (квадратические, кубические) конгруэнтные генераторы;
- генераторы, основанные на регистрах сдвига с линейной или обобщенной обратной связью;
- аддитивные генераторы с запаздываниями на основе последовательностей чисел Фибоначчи;
- высокопроизводительный генератор «вихрь Мерсенна», использующий свойства простых чисел;
- генераторы на базе клеточных автоматов;
- генераторы, основанные на нечеткой логике и т.д.

Особый класс образуют генераторы псевдослучайных последовательностей, выход которых может быть использован в задачах криптографии. Такие генераторы основаны на детерминированных алгоритмах, которые из входной истинно случайной последовательности битов формируют поток битов большей длины, практически независимых друг от друга и подчиняющихся заданному распределению.

3.2 Классификация генераторов псевдослучайных последовательностей

Генераторы псевдослучайных последовательностей относятся к числу важнейших криптографических примитивов. Во многом именно они определяют надежность систем защиты информации, в которых используются. Это делает выбор генераторов очень ответственным шагом. Как уже упоминалось выше, в настоящее время разработано множество различных генераторов псевдослучайных последовательностей. Они основаны на различных принципах работы и заметно отличаются по качеству производимых последовательностей, производительности, сложности реализации.

Существуют разные подходы к классификации генераторов псевдослучайных последовательностей [2-4, 22]. Один из возможных вариантов классификации приведен на рис. 3.1.

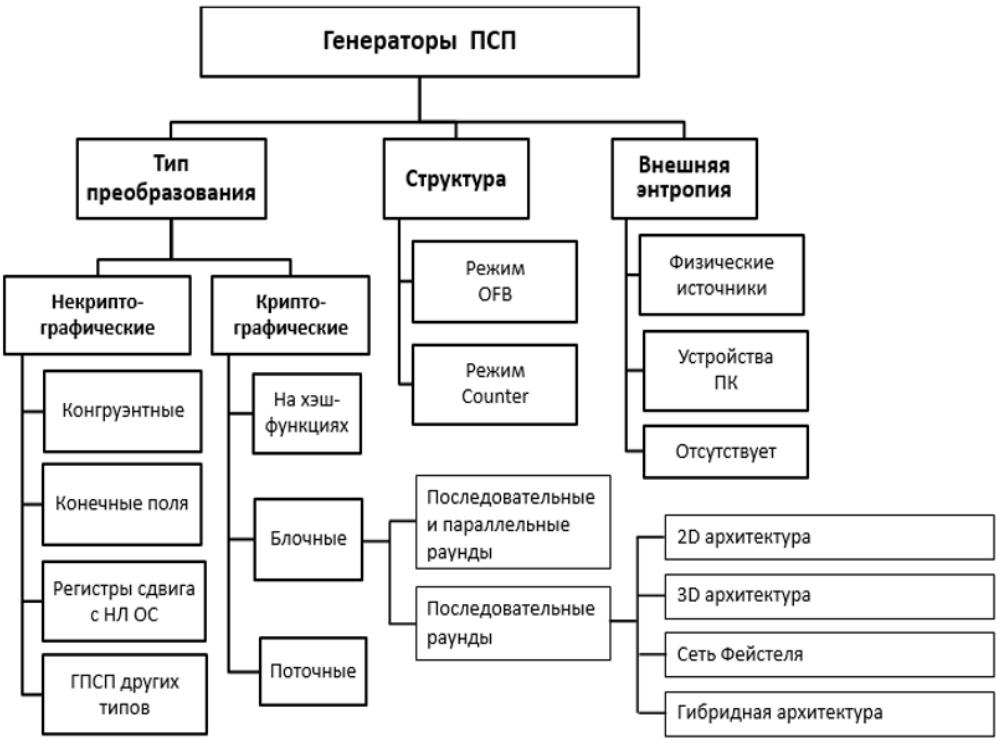

Рисунок 3.1 – Классификация ГПСП.

Любой генератор псевдослучайных последовательностей обладает следующими основными признаками:

- тип нелинейного преобразования;
- структура генератора;
- наличие внешних источников энтропии.

На основе анализа уровня стойкости ко взлому нелинейного преобразования, которое определяет степень и характер изменения входных данных, выделяются так называемые криптографически стойкие генераторы, созданные на основе криптоалгоритмов (блочных и поточных шифров, а также хэш-функций). Этот класс генераторов будет подробно рассмотрен в соответствующем разделе.

С точки зрения реализации различают также программные, аппаратные и комбинированные генераторы ПСП.

Аппаратные генераторы псевдослучайных чисел реализуют различные алгоритмы генерации ПСП, например, на базе ПЛИС (программируемых логических интегральных схем).

ГПСП аппаратного и комбинированного типа применяются для шифрования данных и помехоустойчивого кодирования. Они также входят в состав оборудования для выполнения статистического и имитационного моделирования, испытаний на помехоустойчивость и надежность, в специальные средства измерения, опознавания и тестирования, системы радио- и гидролокации.

3.3 Требования к генераторам псевдослучайных последовательностей

Генераторы псевдослучайных последовательностей должны удовлетворять определенным условиям. Последовательности, полученные с их помощью, должны обладать равномерным распределением (или, по крайней мере, близким к равномерному). Это означает, что в сгенерированной двоичной последовательности количество нулей должно быть примерно равно количеству единиц, содержащихся в последовательности. Кроме того, случайные значения, из которых состоит сгенерированная последовательность, должны быть статистически независимы. Это означает, что не должно быть никаких корреляций как между отдельными битами, так и между группами битов.

Хороший генератор псевдослучайных последовательностей должен быть эффективным. Это означает, что он должен производить последовательности большой длины за максимально короткое время. Такое требование особенно важно для систем, работающих в режиме реального времени. Кроме того, генераторы псевдослучайных последовательностей, применяемые в задачах криптографии, должны быть устойчивы к различным атакам и нестандартным ситуациям. Это означает, что у злоумышленника не должно быть возможности

угадать любой текущий, предшествующий или последующий выход генератора даже при том условии, что ему известна некоторая информация о входных данных генератора, о его внутреннем состоянии или его текущее или более раннее выходное значение.

В отличие от генераторов истинно случайных последовательностей, генератор псевдослучайных последовательностей всегда обладает определенным периодом в силу конечности возможных состояний вычислительной системы. Он может быть сколь угодно большим, но всегда является конечным. После того, как длина последовательности превысила такой период, значения, производимые генератором, начинают повторяться. Поэтому, чтобы избегать появления явных закономерностей в сгенерированной последовательности, необходимо, чтобы ее период был достаточно большим.

3.4 Примеры генераторов псевдослучайных последовательностей

3.4.1 Алгоритм середины квадрата

Исторически одним из первых ГПСП был предложенный в 1946 г. Дж. фон Нейманом алгоритм «середины квадрата», состоящий из следующих шагов:

1. Выбирается n-разрядное начальное случайное рациональное десятичное число x_i (его источником может быть ГСП).
2. На каждом следующем шаге вычисляется квадрат y_i этого числа ($2n$-разрядное число).
3. В качестве очередного случайного n-разрядного числа x_{i+1} выбираются n средних разрядов квадрата предыдущего числа y_i.

Однако полученные таким способом числа оказываются связанными между собой. Кроме того, последовательность этих чисел имеет малый период.

3.4.2 Линейные конгруэнтные генераторы

Целое число a конгруэнтно целому числу b по модулю n, если разность $(a-b)$ делится на n без остатка [7, 9, 23]. При этом остатки от деления на n как a, так и b равны. Конгруэнтность чисел записывается с помощью соотношения

$$a \equiv b \pmod{n}. \qquad (3.1)$$

В генераторах такого типа формирование ПСП выполняется с помощью следующего рекуррентного алгоритма:

$$x_{i+1} = (ax_i + b) \bmod n, \qquad (3.2)$$

т.е. очередной элемент ПСП получается применением операции деления по модулю n к линейно преобразованному предыдущему элементу ПСП x_i. Параметрами линейного конгруэнтного генератора (ЛКГ) являются:

- модуль n;
- множитель (мультипликатор) a;
- приращение (инкремент) b;
- начальное значение (*seed*) x_0.

Параметры a, b, x_0 – целые числа из отрезка $[0,n]$. От их значений существенно зависит качество выходной ПСП. При неудачных сочетаниях параметров можно получить последовательность с малым периодом.

Известно [1], что выходные последовательности ЛКГ имеют так называемую «решетчатую» структуру, т.е. являются легко предсказуемыми и не могут применяться в криптографичесих целях.

Доказано [1], что определяемая формулой (3.1) линейная конгруэнтная последовательность имеет максимальную длину периода n тогда и только тогда, когда:

- b и m являются взаимно простыми числами;
- число $(a-1)$ кратно некоторому простому делителю n;
- число $(a-1)$ кратно 4, если число n кратно 4.

Следует заметить, что выполнение указанных условий обеспечивает максимальный период ПСП, но не гарантирует ее качество.

Для отбрасывания ЛКГ, порождающих заведомо неслучайные последовательности, используют понятие потенциала s линейной конгруэнтной последовательности с максимальным периодом – наименьшего целого числа, такого, что $(a-1)^s=0 \bmod n$ [7]. При $s=\{1; 2\}$ последующие элементы выходной ПСП линейно зависят от предыдущих. Такие последовательности лишены необходимых свойств ПСП. В случае $s=\{3; 4\}$ последовательность более похожа на ПСП, но любой ее элемент по-прежнему существенно зависит от соседних с ним. Считается, что начиная с $s=5$ последовательности ЛКГ могут иметь достаточную степень случайности (что требует подтверждения с помощью различных статистических тестов, т.к. большое значение потенциала – это только необходимое, но не достаточное условие случайности ПСП).

В [1, 4, 9] приводятся наборы параметров для построения ЛКГ максимального периода.

Если инкремент $b=0$, формула (3.2) приобретает вид

$$x_{i+1}=ax_i \bmod n. \qquad (3.3)$$

Частным случаем такого ЛКГ является генератор Парка-Миллера, параметры которого a, b, n соответственно равны 75, 0 и 2^{31}-1=2147483647.

Развитием линейного конгруэнтного генератора являются полиномиальные генераторы (квадратичный, кубический, произвольной степени). Для таких генераторов в формуле (3.2) линейное выражение заменяется полиномом нужной степени, и, соответственно, увеличивается количество параметров.

Линейные конгруэнтные генераторы имеют простой алгоритм и высокую скорость работы. Они неприменимы для задач криптографии, но могут использоваться в различных приложениях, не требующих криптостойкости ПСП (моделирование, лотереи, компьютерные и азартные игры, индустрия развлечений).

3.4.3 Аддитивные генераторы псевдослучайных последовательностей

Дальнейшее развитие идеи ЛКГ состоит в связывании очередного элемента выходной ПСП не с одним, а с двумя предыдущими элементами.

Наиболее известным примером последовательности, в которой очередной элемент зависит от двух предшествующих, является последовательность Фибоначчи. Аддитивный генератор Фибоначчи использует формулу

$$x_{i+1}=(x_i+x_{i-1}) \bmod n. \qquad (3.4)$$

Период такого генератора, как правило, больше, чем у ЛКГ с тем же значением модуля n. Тем не менее, качество полученных таким образом ПСП оказывается недостаточным для использования их в тех случаях, когда требуется высокая степень случайности.

Для преодоления недостатков генератора Фибоначчи был предложен его вариант со следующей формулой общего члена последовательности:

$$x_{i+1}=(x_{i-j}+x_{i-k}) \bmod n, \qquad (3.5)$$

Входящие в индексы формулы (3.5) значения k и j, для которых должно выполняться условие $k>j\geq 1$, называются запаздываниями, а соответствующий генератор называется генератором Фибоначчи с запаздыванием. Для его максимально возможного периода существует оценка [7], обычно формулируемая в виде следующей теоремы:

Если многочлен x^j+x^k+1 является примитивным многочленом над конечным полем (полем Галуа) второго порядка $GF(2)$ (см. раздел 1), то период соответствующего генератором Фибоначчи с запаздыванием равен числу

$$2^{\log_2 n-1}(2^k-1).$$

Один из генераторов этого семейства (Дж.Ж. Митчел, Д.Ф. Мур) определяется формулой

$$x_i=(x_{i-24}+x_{i-55}) \bmod n,\ i\geq 55,$$

где n – четное число, $x_0,\ldots, x_{54}$ – произвольные целые числа.

Известен мультипликативный вариант генератора Фибоначчи (Дж. Марсалья), для которого в формуле (3.5) вместо сложения используется операция умножения. Максимально возможный период такого генератора, реализуемый при уже приводившихся условиях на j и k, равен

$$2^{\log_2 n-3}(2^k-1).$$

Применимые на практике величины запаздываний приводятся, например, в [1, 9]. Следует иметь в виду, что с ростом этих величин при программной реализации генератора увеличивается потребность в памяти, что снижает эффективность алгоритма.

Помимо этого, при тестировании выходной последовательности мультипликативного генератора Фибоначчи (как и ЛКГ) проявляется ее решетчатая структура [4].

Такую же особенность структуры сохраняют последовательности, сгенерированные с помощью более сложно организованного алгоритма, в котором очередной элемент зависит от произвольного количества предшествующих элементов:

$$x_i=(a_1x_{i-1}+\ldots+a_{k-1}x_{i-k+1}+a_k) \bmod n. \qquad (3.6)$$

В некоторых условиях коэффициенты линейной комбинации в правой части формулы (3.6) могут быть определены таким образом, чтобы период последовательности был максимально возможным. Способ нахождения коэффициентов реализуется средствами теории конечных полей и представляет собой сложную математическую задачу.

Аддитивные генераторы имеют высокую производительность, но не являются криптостойкими. Они используются как вспомогательные блоки в составе стойких ГПСП. Также они служат основой некоторых криптоалгоритмов (Fish, Pike, Mush) [9].

3.4.3 Инверсный конгруэнтный генератор

Инверсный конгруэнтный генератор основан на вычислении обратной функции от линейной комбинации предшествующих элементов последовательности:

$$x_{i+1} = \left(ax_i^{-1} + b\right) mod\, n,$$
$$x_{i+1} = 0,\ \ если\, x_i = 0. \qquad (3.7)$$

Существование соответствующих коэффициентов обосновывается в рамках теории конечных полей, а обращение конгруэнтной последовательности является сложной задачей, что затрудняет применение инверсных конгруэнтных генераторов. Качество ПСП таких генераторов подтверждается хорошим прохождением статистических тестов [4]. В частности, графические тесты не выявляют «решетчатой» структуры ПСП.

3.4.4 Генераторы на регистрах сдвига с линейными обратными связями

Генераторы на регистрах сдвига с линейными обратными связями (РСЛОС) играют очень важную роль как собственно в генерации ПСП, так и в различных аспектах защиты данных, таких как контроль целостности данных при их неумышленном искажении (CRC-коды), для самотестирования интегральных схем, при разработке криптоалгоритмов [1-4, 7, 9].

Алгоритм РЛОС использует двоичное представление числа. Линейный регистр сдвига с обратными связями представляет собой совокупность регистра сдвига и функции обратной связи. Регистр сдвига – это упорядоченный набор битов длины n, для которого определена операция сдвига битов на одну и ту же величину влево или вправо. При необходимости извлечь бит содержимое регистра смещается на одну позицию вправо (рис. 3.2).

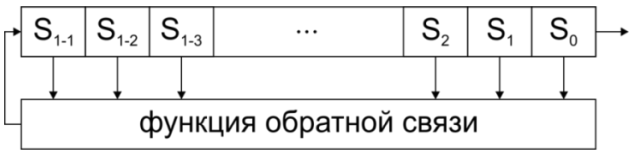

Рисунок 3.2 – Схема работы регистра сдвига с обратной связью.

При генерации очередного бита часть заранее определенных ячеек, называемых «отводами» (*tapped cells*), пропускаются через функцию обратной связи.

В наиболее простом для практической реализации случае функция обратной связи линейна (соответственно линейным является и регистр сдвига). Будем считать, что в качестве такой функции используется операция XOR (исключающее ИЛИ).

Пусть конфигурация отводной последовательности задается последовательностью битов $[c_1,\ldots,c_l]$, в которой отводам соответствуют единицы, а остальным ячейкам – нули. Содержимое регистра сдвига обозначим $[s_{l-1},\ldots,s_0]$.

Шаг генерации очередного бита с помощью регистра сдвига длины l состоит из следующих операций (рис. 3.3):

1. Вычисляется значение функции обратной связи (например, XOR ячеек регистра с коэффициентами $c_1,\ldots,c_l$).
2. Это значение записывается в самую левую ячейку регистра (s_{l-1}).
3. Содержимое всех битов регистра сдвигается на одну позицию вправо.
4. Содержимое крайней правой ячейки регистра (s_0) образует выходное значение генератора.

$$s_j=c_1\cdot s_{j-1}\oplus c_2\cdot s_{j-2}\oplus\ldots\oplus c_l\cdot s_{j-l}. \qquad (3.8)$$

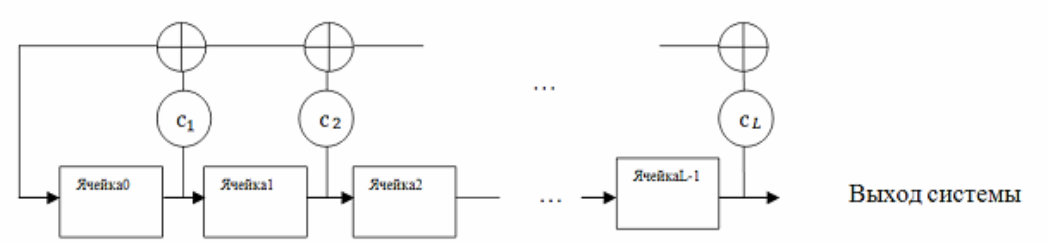

Рисунок 3.3 – Схема работы регистра сдвига с линейной обратной связью.

Получившаяся последовательность обязательно будет иметь период. Максимальное значение различных внутренних состояний регистра, а, значит, период ПСП равняется 2^{l-1}. Последовательность с таким периодом называется M-последовательностью (последовательностью максимального периода).

Математической основой РСЛОС является теория линейных последовательностных машин и теория полей Галуа. В соответствие описанному выше РСЛОС ставится некоторый многочлен, называемый ассоциированным, или образующим, многочленом. Его степень равна длине (разрядности) регистра, а коэффициенты равны элементам отводной последовательности регистра $[c_1,\ldots,c_l]$. Ассоциированный многочлен определяет некоторые важные свойства выходной ПСП [7, 23].

Основные достоинства РСЛОС генераторов [4]:

1. Высокая скорость генерации.
2. Простая реализация как в программном, так и в аппаратном вариантах.
3. Хорошее качество выходной ПСП с точки зрения статистических свойств.
4. Удобство использования в качестве вспомогательных элементов, полезных при решении некоторых специальных задач защиты данных, например, получение ПСП с определенными характеристиками (длина, период, закон распределения).

Криптостойкость генераторов ПСП на базе РСЛОС из-за предсказуемости недостаточна для их использования при защите данных. Однако нелинейные комбинации таких генераторов являются составными частями систем поточного шифрования [9, 23]. Они также находят применение в задачах контроля и диагностики средств вычислительной техники (синтез генераторов псевдослучайных тестовых последовательностей и сигнатурных анализаторов), в системах телекоммуникаций (аппаратная реализация схем помехоустойчивого кодирования, схем скремблирования) и других [24].

Существует специальный вид ГПСП на основе регистров сдвига – с нелинейной обратной связью [4, 7, 9].

3.4.5 Другие варианты генераторов псевдослучайных последовательностей

Разработаны многочисленные модификации генераторов на основе регистров сдвига с линейной обратной связью [4, 9], ориентированные на повышение криптостойкости. Основными характеристиками генераторов, за счет которых возможно достижение этой цели, являются начальное состояние регистра, функция обратной связи, а также алгоритм синхронизации, управляющий сдвигами. Ниже перечислены некоторые наиболее известные генераторы [2, 3, 7, 9].

1. Генератор Джиффи. Использует три РСЛОС, выполняя «перемешивание» двух ПСП.
2. Генератор «стоп-пошел» («Stop-and-Go») (нелинейное объединение двух РСЛОС с управлением 1-м генератором тактовой частотой второго).
3. Генератор (каскад) Голлманна. Три РСЛОС генератора последовательно осуществляют синхронизацию следующего генератора.
4. Регистр сдвига с обратной связью по переносу.
5. Генератор Геффа (нелинейная комбинация РСЛОС).
6. Пороговый генератор (использует результат работы произвольного большого числа РСЛОС).
7. Прореживаемые генераторы (различные варианты управления генератором собственной частотой).

Среди генераторов, не использующих РСЛОС, можно назвать:

1. Вихрь Мерс*е*нна [7].
2. ГПСЧ на базе клеточных автоматов [25].
3. ГПСЧ на базе нечеткой логики [26].
4. ГПСЧ с применением математического аппарата фрактального моделирования [27].
5. Комбинации различных ГПСП.

Национальный стандарт ГОСТ Р ИСО 28640-2012 устанавливает типовые алгоритмы генерации ПСП с различными законами распределения, однако он не касается методов генерации для криптографических приложений.

Вопросы для самоконтроля

1. Что такое псевдослучайная последовательность?
2. Что собой представляет генератор псевдослучайных последовательностей?
3. Назовите основные области применения генераторов псевдослучайных последовательностей.
4. Назовите и охарактеризуйте основные признаки генераторов псевдослучайных последовательностей.
5. Перечислите основные требования, предъявляемые к генераторам псевдослучайных последовательностей.
6. Перечислите несколько наиболее известных генераторов псевдослучайных последовательностей.

Раздел 4. Криптостойкие генераторы псевдослучайных последовательностей

4.1 Общие требования и особенности

Используемые в криптографии и криптоаналитике генераторы псевдослучайных последовательностей должны иметь отвечать более жестким требованиям, чем, например, генераторы для задач моделирования или методов вычислений. Такие генераторы, называемые далее криптографически стойкими (КСГПСП), должны отвечать более строгим критериям:

1. Непредсказуемость результатов работы: при неизвестном ключе/начальном состоянии генератора на основе известной конечной части ПСП невозможно определить как ее последующий элемент, так и предыдущий.
2. Неотличимость статистических свойств генерируемых ПСП от аналогичных свойств истинно случайной последовательности.
3. Большой период последовательности.
4. Эффективная аппаратная и программная реализация.

Непредсказуемость генератора предполагает вычислительную неразрешимость следующих задач:

- определение предыдущего элемента последовательности на основе известного фрагмента ПСП конечной длины (непредсказуемость влево);
- определение последующего элемента последовательности на основе известного фрагмента ПСП конечной длины (непредсказуемость вправо);
- определение ключа по известному фрагменту ПСП конечной длины.

Считается, что непредсказуемый влево генератор является криптостойким [7].

В работе [2] предлагается следующий подход к оценке пригодности ГПСП для задач криптологии. Качественный (иначе – статистически безопасный) ГПСП также должен иметь:

- статистические свойства, близкие к свойствам ИСП;
- нелинейный алгоритм преобразования ключа, обеспечивающий размножение внесенных в исходный текст искажений;
- статистически независимые результирующие ПСП при разных случайных начальных значениях генератора.

Для многих ГПСЧ (конгруэнтных генераторов, генераторов на основе регистров сдвига и других) указанные критерии не выполняются: при хороших статистических свойствах результирующей ПСП алгоритмы могут быть уязвимы для криптоанализа, и в случае компрометации начального состояния генератора становится возможным как получать новые результаты его работы, так и восстанавливать прежние.

4.2 Основные типы криптографически стойких генераторов псевдослучайных последовательностей

4.2.1 Генераторы на основе стойких криптоалгоритмов

В соответствии с приведенной выше классификацией с точки зрения используемого в них преобразования ГПСП можно разделить на некриптографические и криптографические. Преобразованием при этом является функция зашифровывания. Свойство криптостойкости этой функции наследует генератор.

К интересующему нас второму типу криптографических генераторов относятся генераторы на базе блочных и поточных криптоалгоритмов, а также хэш-функций.

Блочные генераторы. Сильной стороной блочных ГПСП является их непредсказуемость, которая обеспечивается нелинейным преобразованием входных данных, реализованным в виде многораундовой структуры. Алгоритмы стойких блочных шифров основаны на преобразованиях подстановки (*substitution*) и перестановки (*permutation*). Многократные повторы этих операций приводят к рассеиванию (*diffusion*) и перемешиванию (*confusion*) битов открытого текста.

В результате рассеивания влияние входных битов и битов ключа распространяется на большое число битов шифротекста. Тем самым скрываются возможные статистические зависимости между битами открытого текста. Перемешивание посредством подстановок усложняет зависимость между ключом и шифротекстом, затрудняя извлечение информации об использованном ключе. В идеальном случае, если бы к каждому элементу открытого текста применялась уникальная подстановка, был бы построен абсолютно криптостойкий шифр. На практике рассеивание и перемешивание применяются совместно, что приводит к появлению лавинного эффекта и обеспечивает высокую криптостойкость шифра.

В качестве подходящего алгоритма шифрования можно использовать криптостойкие блочные шифры, такие как 3DES, AES, ГОСТ 28147-89 (Магма) и ГОСТ 34.12-2015 (Кузнечик), RC6 и другие.

Поточные генераторы. Преимущество поточных генераторов заключается в высокой скорости работы при достаточной степени непредсказуемости.

Поточные алгоритмы шифрования, как правило, используют в качестве ключей заранее сгенерированные ПСП. Их криптостойкость, а также секретность ключевой последовательности, и определяют надежность шифра. Примерами таких шифров могут быть Salsa20, HC-256, Cha-Cha, RC4 и т.д.

Генераторы на основе хеш-функций. Такие генераторы имеют наиболее строго обоснованную непредсказуемость, поскольку она базируется на вычислительной сложности решения математических задач (разложения больших чисел на простые множители или дискретного логарифмирования). Однако генераторы этого типа заметно проигрывают по производительности. Кроме того, использование хеш-функций предполагает предварительное формирование входной последовательности, что не всегда удобно при реализации генератора.

4.2.2 Генераторы, основанные на вычислительно сложных математических задачах

Существуют особые виды КСПСП, например, основанные на вычислительно сложных математических задачах. Алгоритм Блюма–Микали основан на задаче дискретного логарифма, алгоритм Блюм–Блюма–Шуба – на предполагаемой сложности факторизации целых чисел. Последний алгоритм имеет доказанную высокую криптостойкость, но отличается очень низкой скоростью работы и не для всех аппаратных платформ допускает эффективную реализацию [9].

4.2.3 Специальные реализации

Примерами генераторов, использующих специальные реализации, могут служить: алгоритм Ярроу, в котором делается попытка определить энтропию входных данных; псевдоустройства /dev/random и /dev/urandom, реализованные в большинстве POSIX-совместимых ОС; функция CryptGenRandom в составе CryptoAPI компании Microsoft.

Таким образом, можно утверждать, что стойкие алгоритмы блочного и поточного шифрования можно использовать как генераторы псевдослучайных последовательностей.

Непредсказуемость многих криптографических генераторов невозможно доказать строго, так как она базируется на оценках недостаточности различных ресурсов противника для вскрытия используемого генератором криптоалгоритма. Это приводит к необходимости эмпирического исследования качества ГПСП.

Существует иной взгляд на качественный (в отношении криптографии) ГПСП: построение криптографически стойкого генератора можно свести к построению статистически безопасного генератора, который должен удовлетворять следующим требованиям [2-4]:

- по результатам прохождения статистических тестов полученная ПСП не отличается от ИСП;
- для построения генератора используется такое нелинейное преобразование ключа, которое обеспечивает размножение искажений;
- при разных случайных начальных значениях генератор порождает статистически независимые ПСП.

4.3 Применение криптостойких генераторов псевдослучайных последовательностей

Остановимся подробнее на нескольких, наиболее существенных для целей шифрования, вариантах применения ГПСП.

В соответствии с известным принципом Керкгоффса [8] надежность шифрования определяется только секретностью ключа, но не секретностью криптоалгоритмов. При асимметричном шифровании ключ должен отвечать определенным требованиям, связанным с математической основой алгоритма. Для симметричного шифрования секретные ключи являются важнейшей составляющей надежности соответствующих алгоритмов. Идеальным ключом является случайная последовательность битов. На практике вместо такого ключа часто используется качественная ПСП, состоящая из равновероятных битов, не имеющих статистических закономерностей [28].

В асимметричных криптосистемах функция генератора ПСП состоит в том, чтобы, используя короткий секретный ключ k как «зерно» (*seed*), сформировать длинную псевдослучайную последовательность (например, в криптосистемах RSA, ElGamal, ГОСТ 3410).

Длина ключа, обеспечивающая надежность криптоалгоритма, определяется системой шифрования и составляет не менее 128 битов для симметричных систем и не менее 2304 битов для систем с открытым ключом [6, 9].

4.3.1 Формирование ключей для симметричных криптосистем

Качественный ключ, предназначенный для использования в рамках симметричной криптосистемы, представляет собой случайный двоичный набор. Если требуется ключ разрядностью n, в процессе его генерации с одинаковой вероятностью должен получаться любой из 2^n возможных кодов. Генерация ключей для асимметричных криптосистем – процедура более сложная: так, ключи, применяемые в таких системах, должны обладать определенными математическими свойствами. Например, в случае системы RSA модуль шифрования есть произведение двух больших простых чисел.

Для генерации ключевой информации, предназначенной для использования в рамках симметричной криптосистемы, используются следующие методы (в порядке возрастания качества):

1. Программная генерация, предполагающая вычисление очередного псевдослучайного числа как функции текущего времени, последовательности символов, введенных пользователем, особенностей его клавиатурного почерка и т.п.
2. Программная генерация, основанная на моделировании качественного генератора ПСП с равномерным законом распределения.
3. Аппаратная генерация с использованием качественного генератора ПСП.
4. Аппаратная генерация с использованием генераторов случайных последовательностей, построенных на основе физических генераторов шума и качественных генераторов ПСП.

Использование качественного генератора ПСП позволяет при реализации симметричных блочных шифров уменьшить число раундов шифрования, а значит, увеличить быстродействие криптоалгоритма.

4.3.2 Генерация гаммы для синхронных поточных шифров

В синхронном поточном шифре шифрующая последовательность генерируется независимо от потока открытого текста и потока шифротекста. Функционирование генератора гаммы при поточном шифровании иллюстрирует схема на рис. 4.1 [4]. Начальное состояние может быть функцией от ключа *k* и, возможно, от некоторой рандомизирующей переменной. Цель генератора гаммы – развернуть короткий случайный ключ *k* в длинную псевдослучайную последовательность.

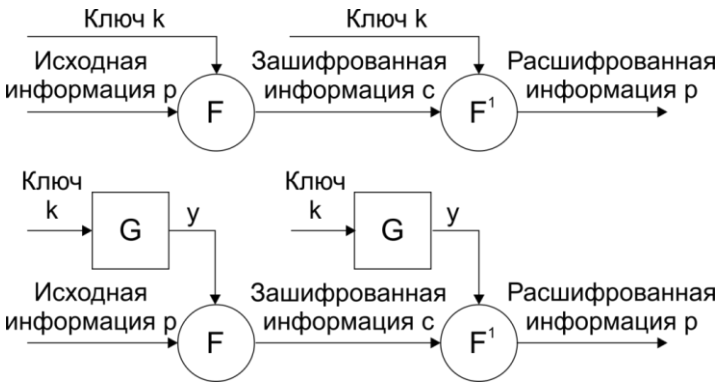

Рисунок 4.1 – Роль ГПСП в процессе шифрования: *а* – абсолютно стойкий шифр; *б* – гаммирование (синхронный поточный шифр); *G* – ГПСП; *F* – линейная (XOR или *mod p*) или нелинейная функция.

4.3.3 Генерация гаммы для самосинхронизирующихся поточных шифров

Так называемые самосинхронизирующиеся или асинхронные поточные шифры, напротив, имеют способность продолжать правильное расшифрование в

том случае, когда шифропоследовательность, генерируемая принимающим шифратором (дешифратором), выпадает из синхронизации с гаммой шифратора передающего. Для таких поточных шифров функция, определяющая следующее состояние криптосистемы, берет в качестве входа фрагмент шифротекста, сгенерированного до этого.

В самосинхронизирующихся поточных шифрах (рис. 4.2) символы открытого текста шифруются с учетом ограниченного числа предшествующих n-символов, которые принимают участие в формировании ключевой последовательности [4]. При этом секретным ключом Z является функция обратной связи генератора ПСП.

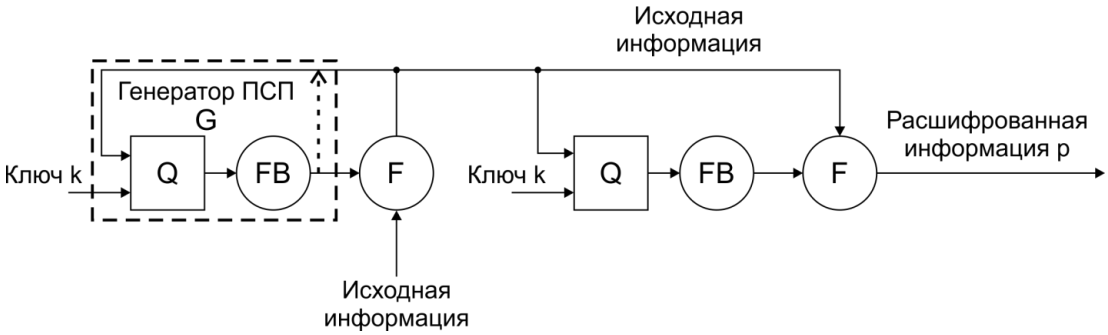

Рисунок 4.2 – Гаммирование с обратной связью (самосинхронизирующееся поточное шифрование) (*FB* – функция обратной связи; *Q* – элементы памяти ГПСП).

Вопросы для самоконтроля.

1. Назовите основные требования, которым должен удовлетворять криптографически стойкий генератор псевдослучайных последовательностей.
2. Перечислите существующие типы криптографически стойких генераторов ПСП.
3. Охарактеризуйте генераторы, базирующиеся на стойких алгоритмах шифрования.
4. Охарактеризуйте генераторы, базирующиеся на вычислительно сложных математических задачах.
5. Охарактеризуйте генераторы, использующие различные специальные реализации.

Раздел 5. Тестирование генераторов псевдослучайных последовательностей

Генераторы псевдослучайных последовательностей могут использоваться для выполнения различных функций. Помимо прочего, они могут использоваться для решения важных задач, в том числе – при защите данных. Поэтому необходима надежная проверка свойств генераторов, таких, как близость выходной последовательности к истинно случайной с точки зрения ее статистических свойств и непредсказуемость выходных значений.

Существует два основных направления анализа псевдослучайных последовательностей:

- **Криптографическое.** Цель этого направления – поиск таких закономерностей исследуемой последовательности, чтобы по ее части можно было восстановить всю последовательность целиком.
- **Статистическое.** Это направление ориентировано на поиск отклонений статистических свойств псевдослучайной последовательности, на основе которых можно предсказывать последующие и предыдущие значения членов последовательности с вероятностью, большей 0,5.

Строгое доказательство непредсказуемости генераторов псевдослучайных последовательностей – сложная проблема, которая до сих пор не решена. В случае криптографически стойких генераторов псевдослучайных последовательностей обычно считается, что для определения следующего бита вырабатываемой последовательности с вероятностью более 0,5 у противника не будет достаточно ресурсов (таких, как времени, вычислительных ресурсов, а также материальных средств).

Существуют теоретические методы исследования псевдослучайных последовательностей, которые используют различные подходы к определению случайности. К ним относятся, например, частотный подход (впервые предложен фон Мизесом), алгоритмический подход Мартин-Лефа, сложностной подход Колмогорова [3, 7]. Однако в настоящее время такие теоретические разработки не дают применимых на практике инструментов оценки степени случайности последовательности.

Существует и другой подход, предполагающий, что некоторый набор статистических свойств псевдослучайной последовательности должен соответствовать аналогичным свойствам истинно случайной последовательности. Проще говоря, псевдослучайная последовательность при прохождении статистических тестов должна вести себя как истинно случайная последовательность. При этом ни один тест не должен обнаруживать в

исследуемой псевдослучайной последовательности какие-либо закономерности, которые бы отличали ее от истинно случайной последовательности.

На практике подтверждение степени случайности результата работы ГПСП означает проверку отсутствия в вырабатываемой последовательности статистических закономерностей и корреляции между последовательностями.

Исторически первые необходимые (но не достаточные) условия статистического сходства периодической двоичной ПСП и ИСП были сформулированы в 1967 г. Они известны как постулаты Голомба [28]:

1. Число единиц и число нулей в каждом периоде ПСП должны отличаться не более чем на единицу.
2. В каждом периоде ПСП половина серий, т.е. подпоследовательностей, состоящих из одинаковых символов, должна иметь длину один, одна четверть – иметь длину два и т.д.; к тому же для каждой из этих длин должно иметься почти одинаковое число серий нулей и единиц.
3. Автокорреляционная функция, являющаяся мерой подобия последовательности и ее сдвига на любое число битов t, не равное периоду, принимает различные значения по мере того, как t проходит все допустимые значения.

Смысл последнего постулата можно трактовать следующим образом: знание предыдущего значения последовательности не позволяет сделать предположение о текущем ее значении.

Однако выполнение этих постулатов не гарантирует случайный характер рассматриваемой последовательности. Для анализа уровня случайности последовательности разработаны и разрабатываются специальные статистические тесты. В настоящее время такие тесты стали основным инструментом оценки качества генераторов псевдослучайных последовательностей.

Такие тесты позволяют:

- оценивать возможность использования ПСП в криптографических приложениях;
- определять генераторы, вырабатывающие последовательности, существенно отличающиеся от СП;
- проверять правильность реализации генераторов ПСП;
- разрабатывать новые генераторы ПСП.

Статистические тесты используются также для оценки качества криптографических примитивов (шифров, хеш-функций) путем проверки криптографических свойств вырабатываемых ими последовательностей (лавинного эффекта изменения выходных данных при искажениях элементов входных данных, корреляции промежуточных и выходных последовательностей).

Впервые это было сделано в ходе известного конкурса по выбору нового криптографического стандарта США Advanced Encryption Standard (AES).

Существуют различные методики тестирования генераторов псевдослучайных последовательностей. Так или иначе, все они базируются на сравнении свойств вырабатываемых последовательностей со свойствами истинно эталонной случайной последовательности. По характеру оценивания и представления тесты, используемые для исследования свойств последовательностей, можно разделить на два основных типа: статистические тесты и графические тесты.

5.1 Статистические тесты

Статистические тесты – это эмпирические тесты для оценки качества ГПСП и выявления их «слабых мест» путем расчета статистических характеристик ПСП и сравнения их с аналогичными характеристиками ИСП [1-4, 7].

Решение таких задач основано на проверке некоторых гипотез относительно свойств ПСП, производимых генераторами. В качестве статистической гипотезы может использоваться произвольное предположение о характере распределения и свойствах случайной величины. Истинность или ложность такого предположения подтверждается или отклоняется с помощью методов математической статистики.

Общий механизм проверки статистических гипотез состоит в следующем. Выдвигаются две гипотезы: нулевая (H_0) и альтернативная (H_1). Предположим, нулевая гипотеза заключается в том, что тестируемая ПСП истинно случайная (с точки зрения конкретного теста), а альтернативная – что ПСП не случайна [3, 7, 11, 12].

Для каждого теста выбирается некоторая вычисляемая функция (статистика), которая сводит свойство случайности тестируемых данных к одному числовому значению (наблюдаемой статистике). Статистика теста представляется как реализация случайной величины. Для того чтобы выполнить оценку прохождения теста, необходимо знать распределение тестовой статистики в предположении, что нулевая гипотеза верна. Часто эту роль играют нормальное распределение и распределение χ^2 (Пирсона) [7, 12].

Статистические тесты соединяют в себе вычислительные процедуры для нахождения статистики исследуемой последовательности и решающее правило проверки, с помощью которого по значениям статистики определяют, принять или отвергнуть нулевую гипотезу:

- если выборочное значение статистики принадлежит критической области, то нулевая гипотеза H_0 отвергается, так как при однократном испытании произошло событие, вероятность которого мала и равна α;

– если выборочное значение статистики попадает в допустимую область, то делается вывод, что данные испытания не противоречат выдвинутой нулевой гипотезе H_0 и она принимается.

Таким образом, алгоритм проверки статистических гипотез состоит из следующих шагов (рис. 5.1):

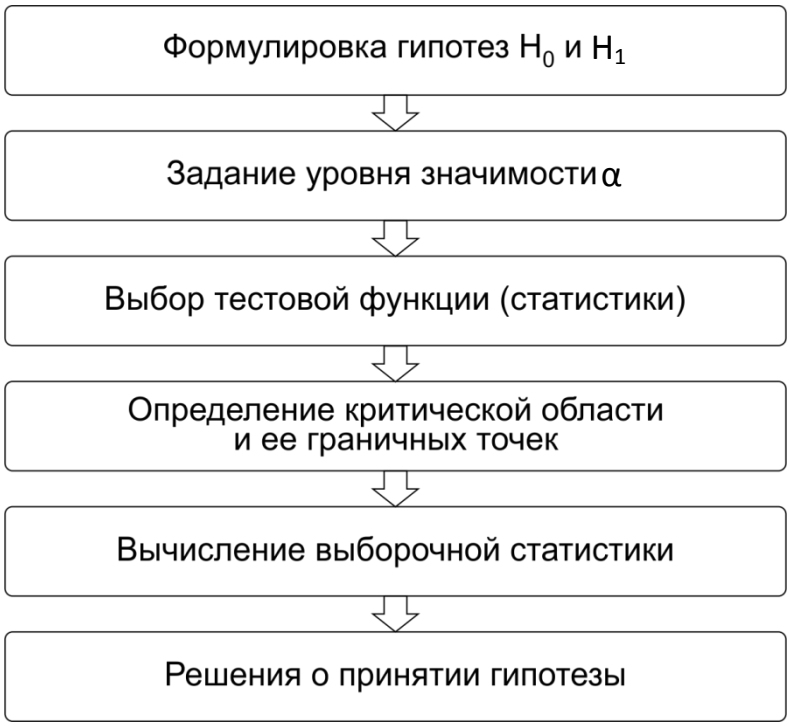

Рисунок 5.1 – Алгоритм проверки статистических гипотез.

Для того чтобы сделать вывод о прохождении теста, проверка этих гипотез выполняется с помощью различных статистических критериев – правил, в соответствии с которыми принимается или отклоняется нулевая гипотеза. Ниже перечислены различные типы таких критериев в порядке нарастания их надежности:

– Пороговое значение. Величина вычисленной статистики сравнивается с некоторым пороговым значением, и, если статистика, например, превосходит его, тест считается пройденным.
– Доверительный интервал. Тест считается пройденным, если величина статистики теста попадает в определенный доверительный интервал, зависящий от принятого уровня значимости.
– Вероятностный подход. Набор значений статистики теста считается набором значений случайной величины с заданным законом распределения.

Последний вариант зарекомендовал себя наиболее эффективным и надежным. Именно он используется во многих пакетах статистических тестов.

При принятии решения о том, был ли пройден тест, возможны два типа ошибок. Возникновение ошибки первого рода означает, что тестируемая псевдослучайная последовательность на самом деле является случайной, но верная нулевая гипотеза H_0 отклоняется. Вероятность такой ошибки равна уровню значимости α, который задается до начала тестирования. Уровень значимости α – это вероятность того, что тестирование покажет неслучайность последовательности, тогда как фактически она является случайной. Соответственно, вероятность принятия правильного решения составляет (1–α). Обычно в практических задачах приемлемым считается значение уровня значимости 0,05. Однако для целей криптографии используют более строгие значения α (как правило, из интервала [0,001; 0,01]).

Ошибка второго рода (β) означает принятие гипотезы о случайности рассматриваемой последовательности, когда последовательность в действительности неслучайна. С точки зрения криптографии такая ошибка более критична. Величина ошибки второго рода определяет мощность критерия – вероятность того, что нулевая гипотеза будет отклонена при верной альтернативной гипотезе. Между двумя этими видами ошибок существует взаимозависимость: чем меньше α, тем больше β, и наоборот.

Статистика теста построена так, что ее меньшие значения соответствуют дефектам псевдослучайной последовательности – отклонениям от истинной случайности.

Обычно для удобства восприятия результатов тестирования вычисленная с помощью эталонного распределения вероятностей тестовая статистика преобразуется в так называемое значение *p-value.* Это значение трактуется как вероятность того, при заданном уровне значимости идеальный генератор случайных последовательностей может произвести последовательность, менее случайную, чем исследуемая. Такое событие тем менее вероятно, чем меньше значение *p-value*. Выполнение условия *p-value*>=α означает успешное прохождение теста.

Важным является тот факт, что для любых статистических тестов, удовлетворяющих нулевой гипотезе, значения *p-value* равномерно распределены на интервале [0; 1). Это означает, что тестируемая произвольным тестом последовательность должна быть равномерно распределена на интервале [0; 1).

Перечислим несколько наиболее известных инструментов статистического тестирования псевдослучайны последовательностей [1, 4, 6]:

- подборка Кендалла и Бабингтон-Смита;
- тесты Д. Кнута;
- тесты DIEHARD (Дж. Марсалья);

- пакет тестов NIST (A. Rukhin и др.);
- пакет TestU01 (П. Л’Экуйе);
- Crypt-XS (Helen Gustafson);
- John Walker (Autodesk, Inc.), ENT;
- Dieharder (Robert G. Brown).

Помимо пакетов, содержащих наборы тестов для многостороннего исследования статистических свойств ГПСП, существуют отдельные тесты, которые направлены на более точный и полный анализ ПСП.

Следует понимать, что тестирование не может заменить криптоанализ. Тем не менее, оно является обязательным этапом анализа стойкости криптографического генератора. В условиях существования большого числа различных статистических тестов, как широко и давно распространенных, так и новых, важен обоснованный выбор, связанный со спецификой решаемых задач защиты информации.

5.2 Графические тесты

Графические тесты используют гистограмму или диаграмму распределения на плоскости элементов последовательности (рис. 5.2), проверку серий, проверку на монотонность, автокорреляционную функцию, профиль линейной сложности и дискретное преобразование Фурье [4, 6]. Графические тесты имеют ту же математическую основу, что и статистические, но уступают статистическим тестам в точности, поскольку ориентированы на поиск явных отклонений псевдослучайной последовательности от эталона. Такие отклонения должны быть хорошо заметны «на глаз» при визуальном восприятии. Графические тесты дают приближенное визуальное представление определенных статистических свойств исследуемой последовательности в виде тех или иных графиков и гистограмм. При использовании графических тестов бывает удобнее работать не с битами, а с числами. По этой причине тестируемую последовательность часто представляют в виде набора, например, 32-битных чисел.

На практике такие тесты оказываются наглядными только в том случае, если свойства псевдослучайной последовательности выражены достаточно отчетливо. Если же ситуация является менее определенной, интерпретация результатов оказывается не вполне однозначной. В итоге трактовка результатов такого тестирования становится все более субъективной, что снижает ценность графических методов тестирования. Таким образом, графические тесты имеет смысл применять на начальных стадиях исследования генераторов псевдослучайных последовательностей для быстрой, поверхностной оценки, которая носит исключительно качественный характер.

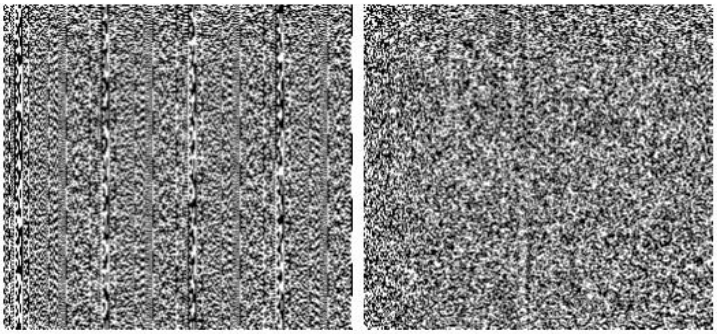

Рисунок 5.2 – Визуализация зависимостей между элементами последовательности (тест «Распределение на плоскости»).

5.3 Пакет статистических тестов NIST STS

Развитие способов генерации требовало проверки статистических свойств получаемых ПСП. При этом предлагались различные варианты такой проверки, и обычно сами тесты и методика их применения разрабатывались для конкретных типов ГПСП. Это затрудняло процесс исследования генераторов и делало невозможным объективное сравнение их качества. Возникла необходимость в стандартном инструменте исследования ГПСП на базе единой методики. В качестве решения этой проблемы Национальный институт стандартов и технологий США (NIST) в 1999 г. выполнил специальные исследовательские работы, результатом которых стал набор из 16 (позднее 15) тестов. В настоящее время этот набор статистических тестов – «NIST STS» (NIST Statistical Test Suite), один из наиболее признанных инструментов для тестирования ПСП, – выполняет функции такого стандарта [30]. Важно отметить, что этот продукт предлагает методику проверки статистических свойств именно криптостойких ГПСП, а также ГСП, т.е. может использоваться применительно к задачам криптозащиты данных. Пакет NIST STS считается наиболее приемлемым с точки зрения строгости оценки свойств ГПСП, эффективным по затратам машинного времени и доступным для использования на различных платформах.

Важный вопрос о степени случайности ПСП не может быть разрешен строго теоретически. Общепринятым является тестирование ПСП для обнаружения возможных недостатков последовательности и проверки ее статистических свойств и криптографической стойкости. Существует и продолжает появляться большое количество инструментов тестирования ПСП.

NIST STS обеспечивает комплексную проверку ПСП в соответствии с обоснованной методикой и предоставляет критерии принятия решения о качестве не только отдельной ПСП, но и ГПСП в целом.

Не исключая возможности применения других тестов, можно утверждать, что пакет NIST STS фактически является основой для исследования

статистической пригодности генераторов псевдослучайных последовательностей, применяемых в области криптографической защиты информации.

Пакет тестов NIST STS состоит из 15 статистических тестов, объединенных общей методикой анализа двоичных генераторов псевдослучайных последовательностей. Дополнительно для демонстрации работы пакета в его состав входят реализации девяти широко известных генераторов псевдослучайных последовательностей (такие, как конгруэнтные генераторы, генератор Блюм – Блюма – Шуба, генератор Микали и другие). Проверка производится для текстовых или битовых последовательностей произвольной длины.

Тесты NIST базируются на проверке гипотезы о случайности исследуемой последовательности (нулевая гипотеза H_0). Следовательно, H_1 – альтернативная гипотеза о неслучайности последовательности. После прохождения каждого теста нулевая гипотеза либо принимается, либо отклоняется. Каждый тест пакета NIST осуществляет сравнение определенной в тесте статистики, вычисленной для тестируемой последовательности, с соответствующим теоретическим значением, которое рассчитывается для выбранного в каждом тесте эталонного распределения случайной величины. В качестве эталонного распределения применяются:

- распределение χ^2 (для подавляющего большинства тестов (10 из 15));
- нормальное распределение (а именно, в двух тестах – стандартное нормальное, еще в трех – одностороннее усеченное нормальное).

В случае нормального распределения производится сравнение тестовой статистики сгенерированной ПСП с ожидаемым значением. Для нахождения статистики значения рассматриваемой ПСП центрируются и нормируются по среднему квадратическому отклонению ($z = \frac{x - \mu}{\sigma}$). Расчет *p-value* выполняется с помощью дополнительной функции ошибок

$$erfc(z) = \frac{2}{\sqrt{\pi}} \int_z^{\infty} e^{-u^2} du . \qquad (5.1)$$

Распределение χ^2 предполагает сравнение степени согласия наблюдаемых частот F_i с соответствующими частотами f_i предполагаемого распределения. Статистикой при этом является величина

$$\chi^2 = \sum_{i=1}^{k} \frac{(F_i - f_i)^2}{f_i} . \qquad (5.2)$$

При этом расчет *p-value* проводится с помощью неполной гамма-функции, формула которой представлена ниже:

$$Q(a,x) \equiv \frac{\Gamma(a,x)}{\Gamma(a)} \equiv \frac{1}{\Gamma(a)} \int_x^\infty e^{-t} t^{a-1} dt,\ \Gamma(z) = \int_0^\infty e^{-t} t^{z-1} dt. \qquad (5.3)$$

Все тесты, входящие в состав пакета NIST STS, параметрические, поэтому при их использовании важен корректный выбор значений необходимых параметров.

Ниже приведем краткое описание тестов, входящих в состав пакета NIST STS.

1. **Частотный (монобитный) тест** (**The Frequency (Monobit) Test**), определяющий нормализованную абсолютную сумму значений элементов последовательности. Тест показывает, не содержится ли в последовательности слишком много нулей или единиц.

2. **Частотный тест внутри блока (Frequency Test within a Block**). Определяет меру согласования количества единиц внутри блока с теоретически ожидаемым значением. Показывает локализованные отклонения частоты появления единиц в блоке от теоретического значения.

3. **Проверка накопленных сумм (The Cumulative Sums (Cusums) Test).** Вычисляется максимальное отклонение накопленной суммы элементов последовательности от начальной точки отсчета. Тест определяет, не слишком ли большое количество единиц или нулей находится в начале или в конце последовательности.

4. **Проверка серий (The Runs Test**). Определяет количество непрерывных серий одинаковых битов на всей длине последовательности. Показывает, есть ли слишком быстрая или слишком медленная перемена серий одинаковых битов в последовательности.

5. **Проверка максимальной длины серии в блоке (Tests for the Longest-Run-of-Ones in a Block**). Определяет меру согласования наблюдаемого значения максимальной длины единичной серии с теоретически ожидаемым значением. Показывает отклонение максимальных длин серий единиц от теоретического закона распределения.

6. **Проверка ранга двоичной матрицы (The Binary Matrix Rank Test**). Определяет меру согласования наблюдаемого значения рангов различного порядка с теоретически ожидаемым. Выявляет зависимость символов в

последовательности по отклонению эмпирического закона распределения значений рангов матрицы от теоретического.

7. **Спектральный тест на основе дискретного преобразования Фурье (The Discrete Fourier Transform (Spectral) Test**). Вычисляет нормализованную разницу между наблюдаемым и ожидаемым количеством частотных компонент, превышающих 95% порогового уровня. Выявляет периодические составляющие в двоичной последовательности.

8. **Проверка перекрывающихся шаблонов (The Overlapping Template Matching Test)**. Определяет меру согласования наблюдаемого количества перекрывающихся шаблонов в последовательности с их теоретическим значением. Показывает наличие в последовательности большого количества *m*-битных серий из единиц.

9. **Универсальный тест Маурера (Maurer's «Universal Statistical»)**. Вычисляет сумму логарифма расстояния между однобитными шаблонами. Указывает на сжимаемость последовательности.

10. **Энтропийный тест (The Approximate Entropy Test)**. Определяет меру согласования наблюдаемого значения энтропии источника с теоретически ожидаемым значением для случайного источника. Характеризует неравномерность распределения *m*-битных слов в последовательности.

11. **Проверка случайных отклонений (The Random Excursions Test)**. Определяется мера согласования наблюдаемого количества посещений при случайном блуждании в заданное состояние внутри цикла с теоретически ожидаемым количеством. Показывает отклонение от теоретического закона распределения посещений в конкретное состояние при случайном блуждании.

12. **Проверка случайных отклонений (вариантный) (The Random Excursions Variant Test**). Определяется общее количество посещений в различные состояния при случайном блуждании. Показывает отклонение от теоретического ожидаемого общего количества посещений при случайном блуждании в различные состояния.

13. **Тест на подпоследовательности (Serial Test**). Определяет меру согласования наблюдаемого количества всех встретившихся вариантов *m*-битных шаблонов с теоретически ожидаемым. Показывает неравномерность распределения *m*-битных слов в последовательности.

14. **Проверка неперекрывающихся шаблонов (The Non-overlapping Template Matching Test**). Определяется мера согласования наблюдаемого количества непериодических шаблонов в последовательности с теоретическим значением. Показывает наличие большого количества заданных непериодических шаблонов в последовательности.

15. **Проверка линейной сложности (The Linear Complexity Test**). Определяется мера согласования наблюдаемого количества событий, заключающихся в появлении фиксированной длины эквивалентного линейного рекуррентного регистра (ЛРР) для заданного блока с теоретически ожидаемым. Показывает наличие отклонения эмпирического распределения длин эквивалентных ЛРР для последовательности фиксированной длины от теоретического закона распределения для случайной последовательности, что указывает на недостаточную сложность тестируемой последовательности.

Каждый тест выполняет расчет определенной статистической характеристики и соответствующих значений вероятности того, что гипотеза о случайности ПСП является правильной.

Тесты различаются по сложности используемых в них алгоритмов и по объему обрабатываемых участков последовательности.

Часть тестов работает с отдельными битами, проверяя наиболее простые характеристики последовательности (количество нулей или единиц, накопленную сумму элементов последовательности, количество серий одинаковых битов в последовательности и т.д.).

Тесты второй группы обрабатывают *m*-битные блоки. Это тесты на встречающиеся пересекающиеся и непересекающиеся шаблоны, универсальный тест Мауэра, тест на подпоследовательности, энтропийный тест.

Самые сложные тесты обрабатывают большие блоки (длиной более 1000 битов): проверка линейной сложности, спектральный тест, проверка ранга двоичной матрицы.

Выводы о прохождении теста для всей выборки делаются по результатам проверки двух следующих условий:

1. Попадание вычисляемой пакетом тестов NIST величины *Pr* – доли последовательностей, сгенерированных с помощью ГПСП, прошедших тест (определяется как отношение количества прошедших тест последовательностей к общему количеству протестированных) – в доверительный интервал

$$\left[(1-\alpha) - 3\sqrt{\frac{\alpha(1-\alpha)}{m}},\ (1-\alpha) + 3\sqrt{\frac{\alpha(1-\alpha)}{m}} \right], \qquad (5.4)$$

где m – размер выборки.

2. Равномерность распределения вероятностей *p-value* на отрезке [0,1].

Для проверки равномерности пакетом NIST вычисляется величина статистики для k=10 интервалов [0; 0,1), [0.1; 0,2), …, [0,9; 1):

$$\chi^2 = \frac{\sum_{i=1}^{k}\left(\nu_i - m/k\right)^2}{m/k}, \qquad (5.5)$$

где ν_i – рассчитанное для каждого интервала количество принадлежащих ему значений *p-value*. В соответствии с критерием χ^2 выполняется проверка того, насколько реальное распределение значений *p-value* близко к теоретическому (равномерному) распределению. Если число прошедших тест последовательностей велико, распределение этой статистики должно приближаться к распределению χ^2 с числом степеней свободы (k-1).

5.4 Пример практического использования пакета NIST STS

При выполнении 15 тестов NIST STS для набора из m последовательностей вычисляются $m\times188$ (с учетом подтестов) значений вероятности p_{ij}-*value*, которые характеризуют прохождение i-й последовательностью j-го теста (i=1,…,m; j=1,…,188).

Если для заранее заданного уровня значимости α выполняется условие p_{ij}-*value*≥α, считается, что i-я последовательность успешно прошла j-й тест. Нарушение этого условия означает появление ошибки первого рода. Для j-го теста программа рассчитывает величину S_{pj} – долю (*proportion*) последовательностей, прошедших его. Окончательным же результатом прогона являются следующие данные, выводимые в файл итогового отчета (*finalAnalysisReport.txt*):

1. 188×10 значений частоты попадания *p-value* в 10 интервалов одинаковой длины, на которые разбивается единичный отрезок;
2. 188 значений *p-value*, определенных в процессе применения критерия χ-квадрат для контроля равномерности распределения вероятностей;
3. 188 значений доли (*proportion*) прошедших каждый тест последовательностей.

Рис. 5.3 иллюстрирует основные этапы работы пакета NIST STS: выбор способа получения тестируемой последовательности (из входного файла или с помощью одного из 8 встроенных генераторов); выбор используемых тестов; задание необходимых параметров тестов. Также при запуске задаются длина

последовательности и количество таких последовательностей. Более подробная информация о работе с пакетом тестов NIST STS приведена в руководстве [30].

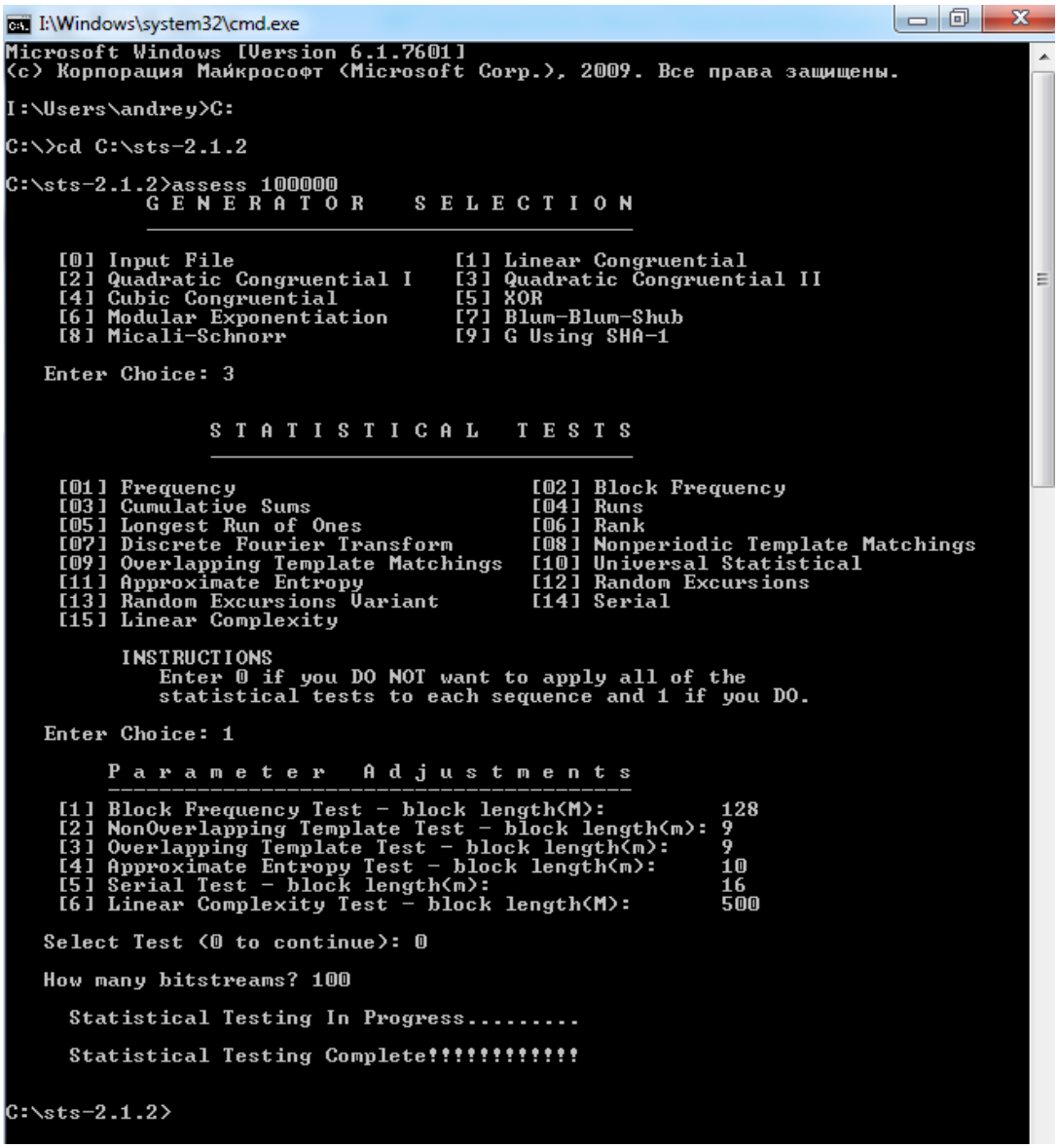

Рисунок 5.3 – Демонстрация работы пакета тестов NIST STS.

Фрагмент файла финального отчета, формируемого пакетом NIST STS, для нескольких первых тестов приведен на рис. 5.4.

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	P-VALUE	PROPORTION	STATISTICAL TEST
6	12	11	7	9	11	11	11	6	16	0,474986	99/100	Frequency
7	7	7	13	14	14	8	16	9	5	0,145326	99/100	BlockFrequency
9	8	8	11	8	7	8	19	10	12	0,262249	100/100	CumulativeSums
8	10	13	10	7	6	11	10	14	11	0,779188	100/100	CumulativeSums
10	10	9	7	8	10	11	5	14	16	0,419021	98/100	Runs
9	14	5	8	14	14	9	12	10	5	0,289667	99/100	LongestRun
8	10	11	11	9	11	11	7	11	11	0,991468	98/100	Rank
11	4	9	14	11	6	14	12	12	7	0,319084	98/100	FFT
13	15	9	3	8	13	7	12	7	13	0,171867	99/100	NonOverlappingTemplate
10	17	12	9	7	11	7	11	8	8	0,514124	99/100	NonOverlappingTemplate
13	10	10	8	4	12	9	13	12	9	0,657933	100/100	NonOverlappingTemplate
15	6	14	13	7	12	7	8	6	12	0,262249	96/100	NonOverlappingTemplate
5	14	11	4	10	15	13	9	11	8	0,224821	99/100	NonOverlappingTemplate
7	7	8	7	14	13	5	15	12	12	0,249284	99/100	NonOverlappingTemplate
12	6	13	18	11	10	3	3	9	15	0,009535	99/100	NonOverlappingTemplate
15	8	11	7	11	10	13	8	8	9	0,759756	98/100	NonOverlappingTemplate
...		...		...		...		...		...		...

Рисунок 5.4 – Фрагмент финального отчета NIST.

На рис. 5.5 и 5.6 представлены диаграммы, характеризующие соответственно попадание доли прошедших каждый тест последовательностей в доверительный интервал [0,96; 1] и равномерность распределения *p-value*.

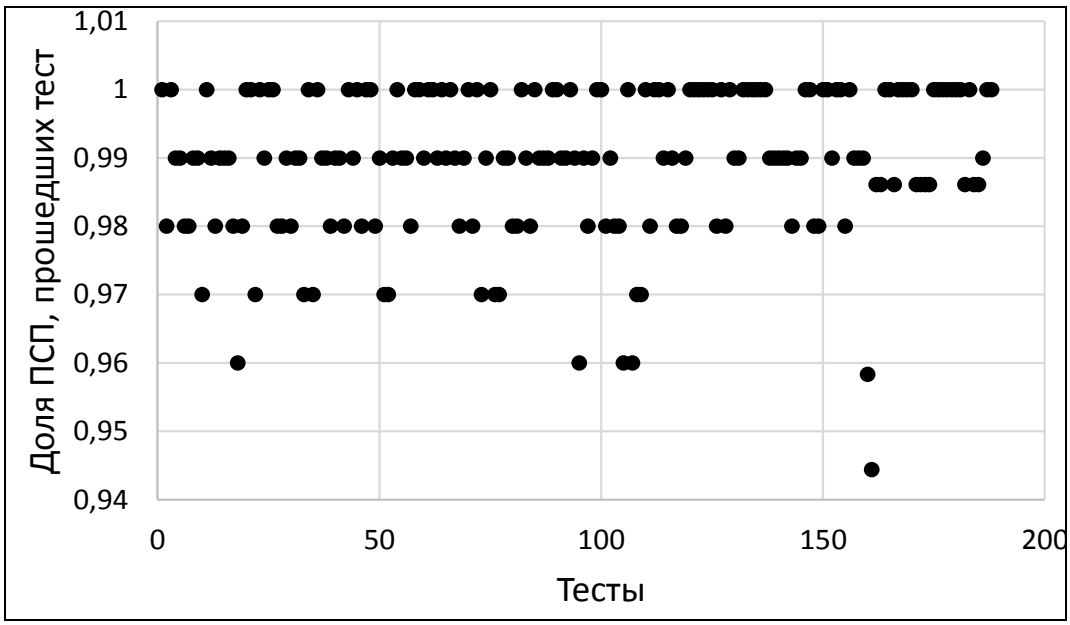

Рисунок 5.5 – Доли прошедших тесты псевдослучайных последовательностей (для всех 188 тестов).

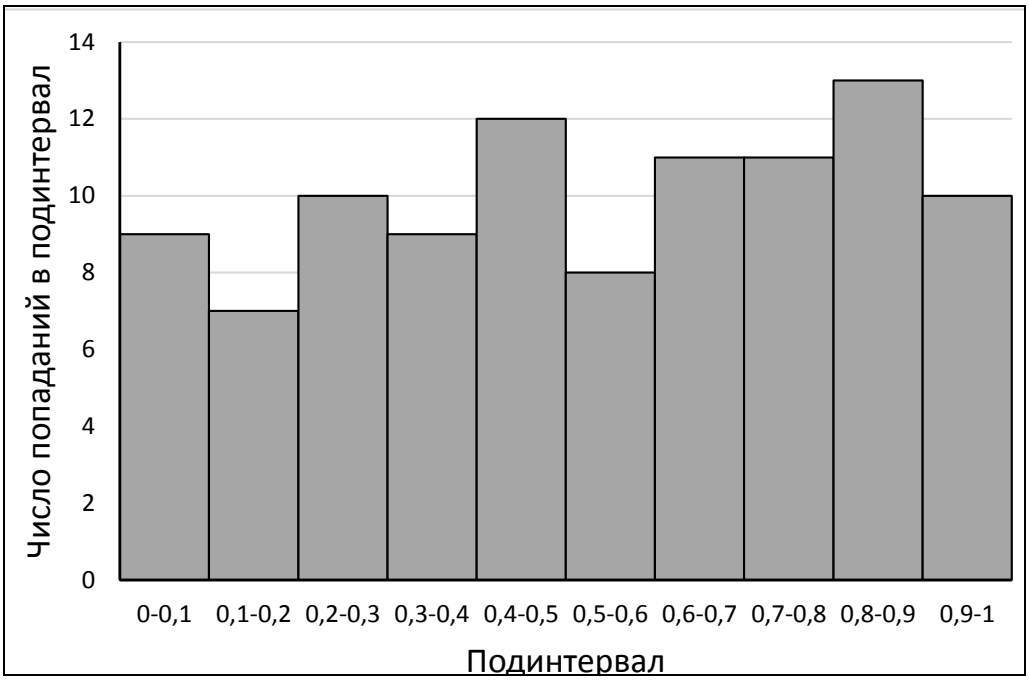

Рисунок 5.6 – Распределение числа прошедших тесты псевдослучайных последовательностей по 10 подинтервалам отрезка [0; 1].

5.5 Другие средства тестирования

Изначально при разработке средств исследования свойств ГПСП авторами тестов предлагались отдельные алгоритмы, часто для определенных генераторов, ориентированные на их особенности, на конкретные виды ПСП.

Постепенно появлялись более совершенные тесты, нацеленные на выявление еще не изученных слабых мест генераторов. Позднее разрабатывались и формировались пакеты («батареи») тестов, которые объединяли различные тесты и позволяли проводить более полное исследование свойств ПСП и их генераторов.

Существенный шаг вперед был сделан по ходу развития техники поточного шифрования. Во время проводившегося в 2004–2008 гг. под эгидой европейского криптологического сообщества ECRYPT конкурса eSTREAM помимо новых алгоритмов шифрования были представлены и новые подходы к тестированию именно КСГПСП. Исследовались взаимосвязи между ключом и генерируемой ПСП, между вектором инициализации и ПСП и анализировалось влияние внутреннего состояния генератора на свойства выходной последовательности. В результате удавалось выявить менее явные свойства генераторов.

Появляются новые тесты, специально разработанные для КСГПСП:

1. «Стопка книг» и дважды адаптивный тест [6, 31].

2. Тесты А. Доганаксоя [32, 33] для анализа свойств синхронных поточных шифров, направленные на обнаружение корреляций между ключом, вектором инициализации, внутренним состоянием генератора и выходной ключевой последовательностью.
3. Топологический двоичный тест [34], дающий точное распределение тестовой статистики и указывающий на сходства и различия с тестами NIST; сильная сторона теста – возможность применения как к длинным, так и к коротким последовательностям.

В [35] разработана методика тестирования псевдослучайных последовательностей на основе используемой в статистической физике модели Изинга.

Происходит совершенствование уже существующих тестов: разрабатываются уточнения математической базы, более эффективные программные реализации или дополнения в виде современного интерфейса. В статье [36] предлагается за счет использования специальных структур данных – таблиц поиска – увеличить скорость выполнения части тестов NIST. Во многих работах описываются дополнения к методике NIST STS. Например, в [37], в частности, уточняется в сторону суживания доверительный интервал для оценки доли *p-value*, прошедших тест.

Существует разработка НИЯУ МИФИ «Система оценки качества (СОК) генераторов псевдослучайных чисел», которая использует как известные (Д. Кнута, ENT, NIST STS, Diehard), так и авторские тесты [38]. Повышение эффективности работы тестов достигается за счет применения технологии nVidia CUDA (организация параллельных вычислений) и метода подсчета числа отсутствующих шаблонов. Названный метод позволяет уменьшить объем используемой памяти, снимает ограничения на размер ПСП, более гибко и экономно по времени реализовать процесс тестирования.

В работе [39] предлагается развитие модели статистического тестирования за счет представления ПСП в виде многомерных массивов. В статье [40] предложена методика, включающая группировку тестов в соответствии с характером тестируемых свойств, а также выбор лучшего из однотипных тестов в группе на основе оценки, учитывающей такие параметры теста, как возможность обработки коротких ПСП и время обработки тестом одной ПСП. В работе [41] предложен комплексный подход к решению различных проблем создания и применения оценочных тестов. В работе отмечаются недостатки существующих методик оценки результатов тестирования и формулируются требования к системе оценки статистической безопасности ГПСЧ. Также предложена структура полнофункциональной системы оценки статистической безопасности ГПСЧ и криптоалгоритмов.

Проверка статистических свойств выходных последовательностей генераторов ПСП является обязательной частью работы по получению ПСП для

целей криптологии. В условиях постоянного появления новых и совершенствования существующих средств тестирования ПСП необходим базовый вариант поверки качества последовательностей, надежный, быстрый, доступный и общепринятый.

Следует заметить, что любые отдельно взятые тесты могут давать ошибочные результаты для конкретного генератора, но в хорошо подобранном наборе тесты дополняют друг друга и взаимно компенсируют присущие им недостатки.

Вопросы для самоконтроля.

1. Что представляют собой статистические тесты?
2. Какие существуют критерии проверки статистических гипотез? Какой из них является самым надежным?
3. Перечислите наиболее известные инструменты статистического тестирования псевдослучайных последовательностей.
4. В чем суть графических тестов? Когда их применение можно считать обоснованным?
5. Охарактеризуйте пакет статистических тестов NIST STS.
6. Расскажите о новых средствах статистического тестирования («стопка книг», тесты Доганаксоя и т.д.).

Приложение 1. Необходимые сведения из теории вероятностей, математической статистики и теории конечных полей

П1.1 Теория вероятностей и математическая статистика

В разделе для справки приведены в алфавитном порядке некоторые использующиеся в пособии понятия и формулы. Предполагается, что читатель имеет базовые знания по этим дисциплинам. Для получения более подробной информации следует обращаться к соответствующей литературе [10-13, 42].

Автокорреляционная функция случайной последовательности $X(t)$ характеризует статистическую зависимость между последовательностью и ее сдвинутой копией от величины сдвига s:

$$\rho(t,s)=\frac{M\left[\left(X(t)-a(t)\right)\left(X(s)-a(s)\right)\right]}{\sigma(t)\sigma(s)},$$

a и σ – математическое ожидание и дисперсия $X(t)$.

Выборка – несколько значений из генеральной совокупности, предназначенные для получения информации о ней.

Гамма-функция – неэлементарная функция, широко используемая в различных областях математики, в том числе в теории вероятностей и математической статистике:

$$\Gamma(x)=\int_0^\infty e^{-t}t^{x-1}dt\,.$$

Генеральная совокупность – для СВ это множество всех рассматриваемых значений этой СВ.

Гистограмма – графическое представление распределения частот для количественного признака, образуемое соприкасающимися прямоугольниками, основаниями которых служат интервалы диапазона наблюдений, а площади пропорциональны частотам попадания в эти интервалы.

Дисперсия СВ – математическое ожидание квадрата центрированной СВ:

$$D(X) = M\left[\left(X - M(X)\right)^2\right].$$

Доверительный интервал (двусторонний) параметра q распределения СВ X с уровнем доверия α – отрезок, центром которого является точечная оценка параметра, в который истинное значение параметра q попадает с заданной доверительной вероятностью p=1-α.

Дополнительная функция ошибок – определяется через функцию ошибок:

$$erfc(x) = 1 - erf(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-u^2} du .$$

Закон распределения вероятностей для СВ устанавливает соответствие между значениями случайной величины и вероятностью их появления.

Критерий χ^2 (критерий согласия Пирсона) – критерий для проверки гипотезы о законе распределения наблюдаемой выборки, при котором нулевая гипотеза использует распределение χ^2.

Критерий согласия распределения – мера соответствия между наблюдаемым распределением и теоретическим распределением.

Неполная гамма-функция – определяется через гамма-функцию:

$$Q(a,x) \equiv \frac{\Gamma(a,x)}{\Gamma(a)} \equiv \frac{1}{\Gamma(a)} \int_x^\infty e^{-t} t^{a-1} dt .$$

Непрерывная случайная величина (НСВ) – это СВ, которая может принимать все значения из некоторого конечного или бесконечного промежутка. Описывается с помощью *функции распределения вероятностей* и *плотности распределения вероятностей*.

Нормальное распределение (Лапласа – Гаусса) – распределение вероятностей непрерывной СВ ***X*** с плотностью распределения

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x-a)^2}{2\sigma^2}} .$$

Параметры a и σ равны математическому ожиданию и дисперсии случайной величины $\boldsymbol{X}$.

Ошибка первого рода – ошибка, состоящая в отбрасывании нулевой гипотезы, поскольку статистика принимает значение, принадлежащее критической области, в то время как эта нулевая гипотеза верна.

Ошибка второго рода – ошибка, состоящая в принятии нулевой гипотезы, поскольку статистика принимает значение, не принадлежащее критической области, в то время как нулевая гипотеза не верна.

Плотность распределения вероятностей определяется как

$$f(x) = F'(x).$$

Позволяет получить представление о близости распределения СВ к одному из известных теоретических распределений.

Равномерное распределение – распределение вероятностей непрерывной СВ с плотностью распределения вероятности, постоянной на отрезке [a,b] и равной нулю вне его:

$$f_X(x) = \begin{cases} \dfrac{1}{b-a}, & x \in [a,b], \\ 0, & x \notin [a,b]. \end{cases}$$

Распределение $\boldsymbol{\chi^2}$ – распределение вероятностей непрерывной СВ $\boldsymbol{X}$ с плотностью распределения

$$f(\chi^2, \nu) = \frac{\left(\chi^2\right)^{\nu/2-1}}{2^{\nu/2}\Gamma(\nu/2)} e^{-\chi^2/2},$$

где ν=1,2,… – число степеней свободы, Γ – гамма-функция.

Случайная величина (СВ) – это переменная, которая может принимать любое из заданного множества значений и характеризуемая *распределением вероятностей* (*законом распределения*).

Среднее квадратическое (**стандартное**) отклонение СВ X – это значение квадратного корня из ее дисперсии:

$$\sigma = \sqrt{D(X)}\,.$$

Статистика – функция от выборочных значений. Это СВ, которая может принимать различные значения для разных выборок. Значение статистики может использоваться для проверки статистических гипотез.

Статистический критерий – статистический метод принятия решений о том, стоит ли отвергнуть нулевую гипотезу в пользу альтернативной или нет.

Столбиковая (столбчатая) диаграмма – графическое представление распределения частот для дискретной случайной величины, образуемое набором столбцов равной ширины, высоты которых пропорциональны частотам.

Уровень значимости – заданное значение верхнего предела вероятности ошибки первого рода.

Функция ошибок (**интеграл вероятности**) – неэлементарная функция, возникающая в теории вероятностей, статистике и теории дифференциальных уравнений в частных производных:

$$erf\left(x\right) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-u^2} du\,.$$

Функция распределения СВ X – функция $F(x)$ на $x \in (-\infty,+\infty)$, значение которой в точке x равно вероятности события $(X<x)$:

$$F\left(x\right) = P\left(X < x\right),\ x\left(-\infty,+\infty\right).$$

Энтропия – в теории информации мера неопределенности случайной величины. Для дискретной СВ, принимающей значения $X_1,\ldots,X_n$ с вероятностями $p_1,\ldots,p_n$, определяется как

$$H\left(A\right) = H\left(p_1, p_2, ..., p_n\right) = -\sum_{k=1}^{n} p_k \log p_k\,.$$

П1.2 Сведения по конечным полям

В современной криптографии (а также в теории чисел, в теории помехоустойчивого кодирования и др.) активно используется математический

аппарат теории конечных полей (полей Галуа), в частности – модулярная арифметика (арифметика остатков).

Приведем кратко основные понятия и сведения, относящиеся к конечным полям, а также несколько простых примеров. Подробные сведения о конечных полях можно найти в [3, 4, 7, 9, 18, 23].

Группа – это множество, для каждой пары элементов которого определена операция сложения (*аддитивная* группа) или умножения (*мультипликативная* группа). Свойства операций подчиняются аксиомам:

1. Группа G является *замкнутой* по операции: $\forall a,b \in G$ элемент $c = a*b$ также принадлежит группе (* – общее обозначение операций).
2. Группа G является *ассоциативной*:

$$\forall a,b,c \in G \quad (a*b)*c = a*(b*c).$$

3. Группа G содержит *нейтральный* элемент e (аналог нуля для сложения или единицы для умножения):

$$a*e = e*a = a.$$

4. Каждый элемент группы имеет *обратный*:

$$\forall a \in G \quad \exists a^{-1}: \; a^{-1}*a = e.$$

Если группа имеет свойство *коммутативности*, она называется *коммутативной* или *абелевой*:

$$\forall a,b \in G \quad a*b = b*a.$$

Примеры групп: множество целых чисел с операцией сложения; множество положительных рациональных чисел с операцией умножения; группа элементов преобразований кубика Рубика; множество из 0 и 1 с операцией сложения по модулю 2.

Понятие группы связано с фундаментальными свойствами симметрии и поэтому оказывается универсальным аппаратом для изучения структур математических объектов самого разного происхождения.

Кольцо – это абелева группа с дополнительными свойствами: определены две операции (сложение и умножение) со следующими аксиомами:

1. Относительно сложения кольцо является *абелевой* группой.
2. Относительно умножения кольцо R является *замкнутым*:

$$\forall a,b \in R \quad c = ab \in R.$$

3. *Дистрибутивность*:

$$\forall a,b,c \in R \quad a(b+c) = ab + ac.$$

4. *Ассоциативность*:

$$\forall a,b,c \in R \quad a(bc) = (ab)c.$$

Примеры колец: множества целых чисел, рациональных чисел, вещественных чисел с операцией умножения.

Поле – это коммутативное ассоциативное кольцо. Относительно операции сложения оно является группой, а относительно умножения группу образуют его ненулевые элементы. Аксиомы поля включают коммутативность и ассоциативность сложения и умножения, существование нулевого, единичного, противоположного и (для ненулевых элементов) обратного элемента, дистрибутивность умножения относительно сложения.

Примеры полей: множества рациональных и вещественных чисел; вычеты (остатки от деления) по модулю.

Следует заметить, что среди полей могут быть и нечисловые, а суть операций – не совпадать с арифметическими сложением и умножением (при тех же обозначениях).

Впервые идеи групп и полей в алгебре использовал французский математик Эварист Галуа (1811–1832).

Далее будут обсуждаться только поля с конечным числом элементов (поля Галуа). Конечное поле, содержащее p элементов, обозначается $\boldsymbol{GF(p)}$. Значение p называется характеристикой поля.

Операция $a \bmod p$ означает *приведение по модулю p* (нахождение остатка от деления a на p).

Запись $a \equiv b \bmod p$ означает, целые числа a и b *сравнимы по модулю p* (разность чисел a и b делится на p без остатка, а остатки от деления этих чисел на p равны). Сравнимые числа рассматриваются как равные в поле, и знак ≡ заменяется обычным знаком =. Иногда говорят, что такие числа являются *конгруэнтными* по модулю p.

Для простого числа p всегда существует поле ***GF*(*p*)**, состоящее из целых чисел от 0 до (p-1), все операции в котором выполняются по модулю p. Такое поле называется *простым*.

Примитивным элементом поля ***GF*(*p*)** называется такой элемент, в виде степеней которого могут быть представлены все элементы поля, кроме нулевого.

Если a – примитивный элемент, то $a^{p-1}=1$. Иногда такой элемент называют *генератором* [9].

В том случае, когда характеристика поля представима в виде p^m, поле Галуа ***GF*(p^m)** с *основанием p степени m* образует полиномы степени m-1:

$$a_0 + a_1 x + a_2 x^2 + \ldots + a_{m-1} x^{m-1}$$

коэффициенты которых являются элементами простого поля ***GF*(*p*)**. Такое поле называется *расширенным*.

Следует заметить, что именно поля полиномов играют важную роль в таких областях, как криптография, кодирование.

Полином над полем ***GF*(p^m)** называется *неприводимым*, если его нельзя представить как произведение двух полиномов над тем же полем, имеющих меньшие степени. Неприводимый полином – это аналог простого числа.

Для поля ***GF*(p^m)** также вводится понятие примитивного полинома, путем возведения в степень которого можно получить все ненулевые элементы этого поля.

Поле Галуа ***GF*(p^m)** можно построить, если задать его характеристику (основание) p, степень m и выбрать так называемый *порождающий* (*образующий*) полином $G(x)$. Порождающий полином должен быть неприводимым. Все арифметические операции выполняются в поле ***GF*(p^m)** по модулю порождающего полинома $G(x)$.

Сложение полиномов в поле Галуа происходит аналогично обычному сложению полиномов: суммируются (по модулю p) коэффициенты слагаемых с одинаковыми показателями степени.

Рассмотрим несколько примеров, иллюстрирующих построение полей Галуа и реализацию в них некоторых арифметических операций.

1. Для простого поля ***GF*(3)** результаты операций сложения и умножения представлены в табличной форме:

+	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

*	0	1	2
0	0	0	0
1	0	1	2
2	0	2	1

2. Примитивным элементом для простого поля ***GF*(7)** является 3:

$$3^0=1,\ 3^1=3,\ 3^2=2,\ 3^3=6,\ 3^4=4,\ 3^5=5.$$

3. Для расширенного поля $\boldsymbol{GF(2^2)}$ перечислим составляющие его элементы. Это четыре неприводимых полинома: 0, 1, x, x+1.
4. Приведем порождающие полиномы $G(x)$ для четырех первых полей по основанию 2 $\boldsymbol{GF(2^m)}$:

m=1: $1+x$

m=2: $1+x+x^2$

m=3: $1+x+x^3$; $1+x^2+x^3$

m=4: $1+x+x^4$; $1+x^3+x^4$

5. Перечислим элементы поля Галуа $\boldsymbol{GF(2^3)}$, построенного с помощью порождающего полинома $p(x)=x^3+x+1$:

$$0,\ 1,\ x,\ x^2,\ 1+x,\ x+x^2,\ 1+x+x^2,\ 1+x^2.$$

В области информационных технологий удобно использовать поля Галуа с основанием 2. В этом случае простое поле состоит из элементов 0 и 1. Операция сложения при этом – это операция XOR (исключающее ИЛИ).

Литература

1. Кнут Д.Э. Искусство программирования. Том 2. Получисленные алгоритмы. – М.: Вильямс, 2007. – 832 с.
2. Иванов М.А., Михайлов Д.М., Чугунков И.В. и др. Стохастические методы и средства защиты информации в компьютерных системах и сетях. – М.: Кудиц-Пресс, 2009. – 512 с.
3. Иванов М.А., Чугунков И.В. Криптографические методы защиты информации в компьютерных системах и сетях. – М.: НИЯУ МИФИ, 2012. – 400 с.
4. Иванов М.А., Чугунков И.В. Теория, применение и оценка качества генераторов псевдослучайных последовательностей. – М.: КУДИЦ-ОБРАЗ, 2003. – 240 с.
5. Осмоловский С.А. Стохастическая информатика: инновации в информационных системах. – М.: Гор. Линия–Телеком, 2012. – 320 с.
6. Рябко Б.Я., Фионов А.Н., Шокин Ю.И. Криптография и стеганография в информационных технологиях. Новосибирск: Наука, 2015. – 239 с.
7. Слеповичев И.И. Генераторы псевдослучайных чисел. – Саратов: СГУ, 2017. – 118 с.
8. Фергюссон Н., Шнайер Б. Практическая криптография. – М.: Изд. дом «Вильямс», 2005. – 424 с.
9. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. – М.: Триумф, 2002. – 816 с.
10. Гнеденко Б.В. Курс теории вероятностей: Учеб. 7-е изд., испр. М.: Эди-ториал УРСС, 2001. 320 с.
11. Буре В.М., Парилина Е.М. Теория вероятностей и математическая статистика. – С.-Петербург: Лань, 2013. – 416 с.
12. Ивановский Р.И. Теория вероятностей и математическая статистика. Основы, прикладные аспекты с примерами и задачами в среде Mathcad. – С.-Петербург: БХВ-Петербург, 2008.– 528 с.
13. Вентцель Е.С. Теория вероятностей. М.: Высшая школа, 2001. – 575 с.
14. Herrero-Collantes M., Garcia-Escartin J.C. Quantum Random Number Generators // Review of Modern Physics. – 2016. – Vol. 89. – No. 1. – Pp. 1–54.
15. Stipcevic M., Koc C.K. True random number generators. Open Problems in Mathematics and Computational Science. – Springer, 2014. – Pp. 275–315.
16. Шеннон К. Математическая теория связи // Работы по теории связи и кибернетике. – М.: Изд-во иностр. лит., 1963. – С. 243–332.
17. Bagini V., Bucci M. A design of reliable true random number generator for cryptographic applications // In Koç C.K., Paar C. (eds) Cryptographic Hardware and Embedded Systems. – Springer, Berlin, Heidelberg, 1999. – Vol. 1717. – P. 204–218.
18. Нильсен М., Чанг И. Квантовые вычисления и квантовая информация. М.: Мир, 2006. – 824 с.

19. Жданов О.Н., Золотарев В.В. Методы и средства криптографической защиты информации: Учебное пособие. – Красноярск, 2007. – 217 с.
20. Чивилихин С.А. Квантовая информатика. Учебное пособие. – С.-Петербург: СПбГУ ИТМО, 2009. – 80 с.
21. Sunar B., Martin W.J. and Stinson D.R. A provably secure true random number generator with built-in tolerance to active attacks // IEEE Trans. on Computers. – 2007. – No. 56 (1). – Pp. 109–119.
22. Иванов М.А., Скитев А.А., Стариковский А.В. Классификация генераторов псевдослучайных чисел, ориентированных на решение задач защиты информации // REDS: Телекоммуникационные устройства и системы. – 2017. – Т. 7. – № 4. – С. 484–487.
23. Смарт Н. Криптография. – М.: Техносфера, 2005. – 528 с.
24. Иванюк А.А. Проектирование конфигурируемого сдвигового регистра с линейной обратной связью // Информатика. – 2013. –№ 3. – С. 82–92.
25. Жуков А.Е. Клеточные автоматы в криптографии. Часть 2 // Вопросы кибербезопасности. – 2015. – № 4 (22). – С. 47–66.
26. Аникин И.В., Альнаджар Х.Х. Генератор псевдослучайных чисел, построенный на нечеткой логике // Информация и безопасность. – 2015. – № 3. – С. 376–379.
27. Пителинский К.В. Роль коммуникаций в информационном обществе и фрактальные алгоритмы шифрования данных / К.В. Пителинский, А.В. Синьковский // Вопросы защиты информации. – 2005. – № 4. – С. 15–17.
28. Жданов О.Н. Методика выбора ключевой информации для алгоритма блочного шифрования. – М.: ИНФРА-М, 2014. – 88 с.
29. Menezes A., Oorschot P. van, Vanstone S. Handbook of Applied Cryptography. – CRC-Press, 1996. – 816 p.
30. Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications [Электронный ресурс] / A. Rukhin, J. Soto, J. Nechvatal et al. – Режим доступа:
http://csrc.nist.gov/publications/nistpubs/800-22-rev1a/SP800-22rev1a.pdf
31. Монарев В.А. Дважды адаптивный тест для проверки гипотезы о равномерном распределении // Вестник СибГУТИ. – 2015. – № 4. – С. 99–104.
32. Doganaksoy A., Sulak F., Uguz M., Seker O., Akcengiz Z. Mutual correlation of NIST statistical randomness tests and comparison of their sensitivities on transformed sequences // Turkish Journal of Electrical Engineering and Computer Sciences. – 2017. –No. 1. – Pp. 655–665.
33. Doganaksoy A., Sulak F., Uguz M., Seker O., Akcengiz Z. New Statistical Randomness Tests Based on Length of Runs // Mathematical Problems in Engineering. – Vol. 2015. – Article ID 626408.
34. Alcover P., Guillamon A., and Ruiz M. A new randomness test for bit sequences // Informatica. – 2013. – No. 24. – Pp. 339–356.

35. Шерешик А.Ю. Разработка алгоритмов тестирования псевдослучайных последовательностей и хеширования данных на основе модели Изинга : автореф. дис. ... канд. технич. наук: 05.13.19. – Омск, 2013. – 19 с.

36. Касторнов К.А., Панкратова Е.А. Применение таблиц поиска для увеличения скорости выполнения статистических тестов NIST / Сб. трудов XIII Международной научно-технической конференции студентов и аспирантов.– Смоленск: Изд-во Универсум, 2016. – С. 280–283.

37. Sys M., Riha Z., Matyas V. On the interpretation of results from the NIST Statistical Test Suite // Romanian Journal of information science and technology. – 2015. – Vol. 18. – No. 1. – Pp. 18–32.

38. Прокофьев А.О., Чугунков И.В., Матрюхина Е.А., Гриднева Е.А. Вопросы построения программных систем оценки качества стохастических алгоритмов // Современные информационные технологии и ИТ-образование. – 2016. – Т. 12. № 3–1. – С. 169–178.

39. Коренева А.М., Фомичев В.М. Статистическое тестирование псевдослучайных последовательностей // Безопасность информационных технологий. – 2016. – № 2. – С. 36–42.

40. Мордашов А.С. Статистическое тестирование российского стандарта функции хэширования ГОСТ 34.11-2012 («СТРИБОГ») // Вопросы кибербезопасности. – 2015. – № 3 (11). – С. 56–59.

41. Прокофьев А.О., Чугунков И.В., Матрюхина Е.А., Гриднева Е.А. Вопросы построения программных систем оценки качества стохастических алгоритмов // Современные информационные технологии и ИТ-образование. – 2016. – Т. 12. № 3–1. – С. 169–178.

42. Блинова И.В., Попов И.Ю. Теория информации: Учебно-методическое пособие. – Санкт-Петербург: Университет ИТМО, 2018. – 84 с.

Будько Марина Борисовна
Будько Михаил Юрьевич
Гирик Алексей Валерьевич
Грозов Владимир Андреевич

Методы генерации и тестирования случайных последовательностей
Учебное пособие

В авторской редакции
Редакционно-издательский отдел Университета ИТМО
Зав. РИО Н. Ф. Гусарова
Подписано к печати
Заказ №
Отпечатано на ризографе

Редакционно-издательский отдел
Университета ИТМО
197101, Санкт-Петербург, Кронверский пр., 49